FINAL DEGREE PROJECT

ANALYZING AND MODELING ENERGY HARVESTING WIRELESS SENSOR NETWORKS

Author:
Joan VENTURA JAUME

Advisor:
Dr. Kaushik CHOWDHURY

ESCOLA TECNICA SUPERIOR D'ENGINYERS DE TELECOMUNICACIO DE BARCELONA

June 22, 2011

To you, you, and everybody

Acknowledgements

I would like to dedicate some lines to the people who not only made possible this thesis but that also contributed to the unforgettable experience during these years in Technical University of Catalonia and Northeastern University. First of all, I would like to thank Professor Kaushik Chowdhury for offering me this opportunity to come to Boston and work in this lab. Thank you for your motivational and inspiring words.

Secondly, I will be always grateful to the faculties in UPC for the work done to bring us to the United States, giving us the chance to discover these great universities and work with the best. Through them, we obtained the economical support from Obra Social Bancaja and ETSETB. I sincerely appreciate all their support.

I want also to thank my friend Jaume, for standing my bad days and also the good ones; as well as Laura, Burak and Rafa for all the time we have spend together during these months. In the same level of gratitude, all the members of this lab: Rahman, Prusayon, Jiaqi and Ning for your help, for being there everyday sharing great discussions.

I deeply appreciate the help during these years in Barcelona of several classmates with who I had the pleasure to work side by side. A part of this hard work is also yours.

To my friends in Boston, thank you for making this year one of the best experiences in my life; I am sure you will success in science, engineering and all aspects. To my colleagues within BEST and Penyafort, thank you for making life in the university enjoyable, fun and worthy.

Last but not least, I would like to dedicate these last words to my parents, family and other friends for their understanding, support and love. Thank you for believing more in me than I do. I owe it all to you.

Abstract

Energy harvesting is envisaged as an enabling technology to meet the growing energy demands of the 21st century. The current state of the art allows tapping into several physical and naturally existing sources, such as solar, wind, vibration, RF scavenging, among others. However, there is a lack of theoretical models that can predict future consumption and residual availability of energy in a sensor node equipped with multiple boards that harvest a particular source or even simultaneously operate on different types of sources.

In this thesis, we propose MAKERS, a Markov model based method to capture the energy states of such a multi-harvesting board sensor. MAKERS allows detailed predictions of the (i) probability of a node failing to detect an event owing to lack of energy, as well as the (ii) average time before this happens. Compared to previous work in this area, our model has a simpler closed form expression, it is not limited to a sensor having a single-harvesting board, and finally, it considers a more realistic harvesting model. Monte-Carlo simulation results reveal a close fit between the closed form expression in MAKERS and observed values, thereby verifying the accuracy of our approach.

We later revise the model in order to relax the first constraints and move into a more realistic environment. Using some of this modifications, we conducted a set of experiments to analyze the proposed model. Results measuring the average time before running out of energy in real cases show a good agreement with theoretical predictions.

Finally, an extension of the MAKERS model based on the Markov Decision Processes framework, allowing the node to decide which action is the best in order to optimize the number of successful events while increasing its lifetime.

The work presented here pretends to be the first step on modeling multiple-source energy harvesting nodes within the wireless sensor networks field.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This thesis is outlined as follows: In Chapter 1, we provide the reader some background in Wireless Sensor Networks and Energy Harvesting. In Chapter 2 we develop the general case of the MAKERS model. We discuss later, in Chapter 3, some modifications to the model, relaxing some of the first assumptions considered. In Chapter 4, we take a look to the performance evaluation of the model comparing it to real cases. In Chapter 5, we present an opportunistic transmission model for energy-constrained nodes based on MAKERS and build over the Markov Decision Process framework. Finally, in Chapter 6, we conclude our work.

1.1 Wireless Sensor Networks

1.1.1 Introduction

A wireless sensor network (WSN) is a group of spatially distributed sensors that work in an autonomous manner to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants. Sensor nodes collaborate between them to send their data through the network to a main destination, e.g. a sink or a base station. The more modern networks are bi-directional, enabling also to control the activity of the sensors. The ideal wireless sensor network is scaleable, consumes little energy, smart and software programmable, capable of fast data acquisition, reliable and accurate over the long term, costs little to purchase and install, and requires no real maintenance; [1] [20].

WSNs have gained plenty attention both in industry and research in recent years, particularly with the proliferation of Micro-Electro-Mechanical Systems (MEMS) technology which has facilitated the development of smart sensors. Although their limited processing and computing resources, they are inexpensive compared to traditional sensors, which has aided to their worldwide adoption. WSN applications can be classified into two categories: monitoring and tracking. Monitoring applications include indoor/outdoor environmental monitoring, health or wellness monitoring, power monitoring, inventory location monitoring, factory and process automation, and seismic and structural monitoring. Tracking

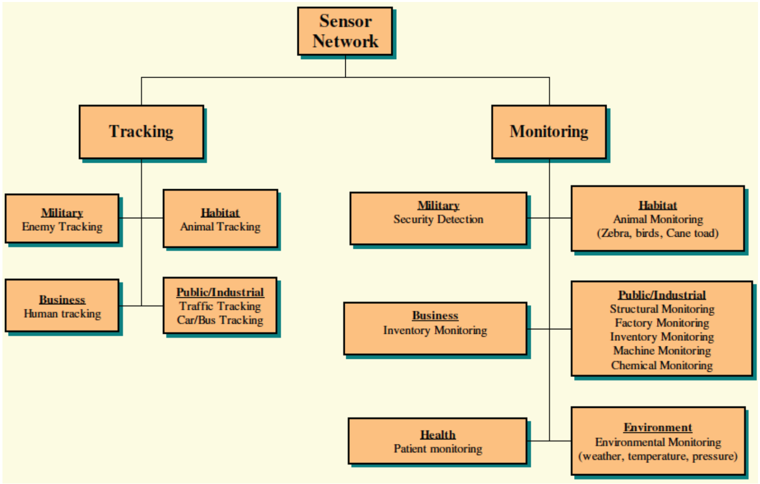


Figure 1.1: Overview of Applications of WSN

applications include tracking objects, animals, humans, and vehicles. Figure 1.1 shows an overview of the applications of WSN, classified into monitoring and tracking applications.

Sensor nodes have typically several parts: a radio transceiver/receiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or a supercapacitor. Additionally, in the last decade a lot of effort has been done to implement energy harvesting capabilities to such sensors, as we will discuss later in Section . Nodes might vary in size from that of a shoebox down to the size of a grain of dust, although functioning motes of this microscopic dimensions have yet to be developed. The cost of sensor nodes is similarly variable, ranging from hundreds of dollars to a few cents, depending on their complexity. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth.

The number of nodes in a WSN can vary from a few to several hundreds or even thousands. Each node is connected at least to another sensor within the network and usually to several. Sensors are normally reachable from any other sensor, using direct communication or sending data through many intermediate sensors.

The position of the sensor nodes does not need to be pre-determined. Random deployment may be allowed in inaccessible terrains or disaster relief operations. However, on the other hand, this implies sensor networks to implement protocols and algorithms with self-organizing capabilities to deal with these deployment difficulties. Another unique feature of sensor networks is the cooperative effort of sensor nodes. Instead of sending the raw data to the nodes responsible for the fusion, sensor nodes use their processing abilities to locally carry out simple computations and transmit only the required and partially processed data.

Although the apparent similarities between wireless sensor networks and traditional wireless ad hoc networks, many protocols and algorithms proposed for these ones are not well suited for the unique features and application requirements of sensor networks. The number of nodes in WSN can be much higher and usually densely deployed. Therefore, they may not have a global ID in order to prevent overhead. Sensor nodes are likely to fail, making the topology of the network change frequently, and mainly use broadcast communication whereas most ad hoc networks are based on point-to-point communications.

1.1.2 Applications

Several different kinds of sensor may form a network such as seismic, low sampling rate magnetic, thermal, visual, infrared, acoustic or radar. This variety of sensor are able to monitor a wide range of parameters as the ones that follow: temperature, humidity, movement, lightning condition, pressure, soil makeup, noise levels, the presence or absence of objects, mechanical stress levels on attached objects; speed, direction and size of an object... Sensor nodes can operate continuously sensing, detecting events, sensing locations and controlling local actuators. WSN applications can be divided in the following categories:

- Military: Monitoring friendly forces, equipment and ammunition; battlefield surveillance; reconnaissance of opposing forces and terrain; targeting; battle damage assessment; and nuclear, biological and chemical attack detection and reconnaissance.

- Environmental: Tracking animals' movements; monitoring environmental conditions that affect crops and livestock; irrigation; large-scale Earth monitoring and planetary exploration; chemical/biological detection; precision agriculture; biological, Earth, and environmental monitoring in marine, soil, and atmospheric contexts; forest fire detection; meteorological or geophysical research; flood detection; bio-complexity mapping of the environment; and pollution study.

- Health: Providing interfaces for the disabled; integrated patient monitoring; diagnostics; drug administration in hospitals; monitoring the movements and internal processes of small animals; monitoring human physiological data; and tracking and monitoring doctors and patients inside/outside a hospital.

- Smart homes: Home appliances automation; smart environment, furniture and appliances that can communicate with each other and a room server that self-organize and adapt their systems based on control theory models.

- Other applications: Monitoring material fatigue; building virtual keyboards; managing inventory; monitoring product quality; constructing smart office spaces; environmental control in office buildings; robot control and guidance in automatic manufacturing environments; interactive toys; interactive museums; factory process control and automation; monitoring disaster area; smart structures; machine diagnosis; transportation; factory instrumentation; local control of actuators; detecting and monitoring car thefts; and vehicle tracking and detection.

1.1.3 Network issues and challenges

To perform the listed applications for WSNs, interaction of the network's elements becomes necessary. The common participants in this interaction are: (i) *sources* that generate the data, (ii) *intermediate nodes* that make additional processing or forwarding of the information, and, finally, (iii) *sinks* where all this data is received. In some applications, these sinks are part of the networks, whereas in others they are external elements that enquire information from the network. The most frequent patters of interaction are:

- Event detection: Source(s) should report to the sink(s) once they have data events matching their tasks. Generally, more than one sensor node is required for most of the applications.
- Function approximation: When in the sink(s) it is required to have an approximation mapping of the area defined for the application. Here, sensor nodes are used to approximate a function of location that estimates the physical value changes from one point to another.
- Periodic measurement: Source(s) can be scheduled to report measured values periodically to a defined interval, according to the application requirements.
- Tracking: In some applications, the task of the WSN is to obtain information (e.g. velocity, direction, size) of an object of interest that presents a mobile behavior. In order to generate meaningful information, various nodes must interact.

Wireless sensor networks demand certain attributes especially related to the characteristic requirements and mechanisms of such systems that we just described. The realization of these characteristics is the major challenge foreseen to WSNs. In figure 1.2, we observe some of the major issues and challenges of WSN. The most important ones are:

- Type of service: WSN should offer the user meaningful information about the object or environment of interest.
- Quality of Service (QoS): It is a metric of the service quality a WSN is offering to its users or applications. The level of QoS is defined by a set of attributes like delay, jitter, bandwidth, packet loss. It is directly related with the type of service it is providing. In the specific case of WSNs, QoS relies in the amount and quality of information extracted from the sinks.
- Fault tolerance: It is essential that the network is capable to deal with failure of sensor nodes. One way to achieve this characteristic requirement is to consider a redundant deployment of sensor nodes. However, this is not always possible.
- Lifetime: It is the time for which the network is operational. It is expected that the WSN can operate at least for the time it was designed for. Nevertheless, lifetime definition depends on the

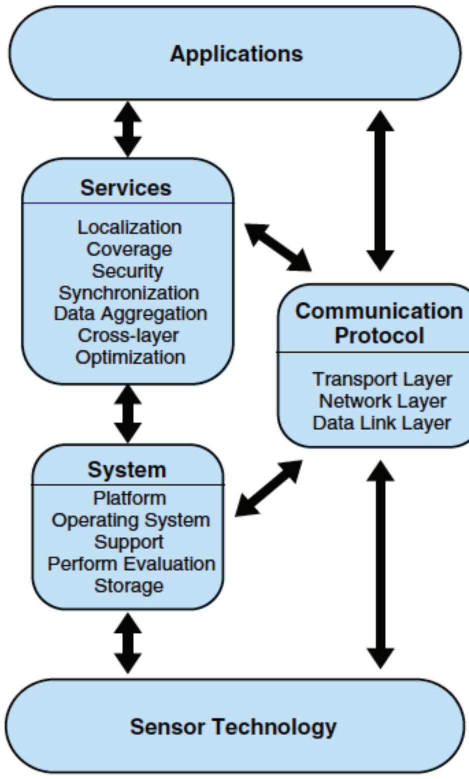


Figure 1.2: Overview of the issues and challenges in WSN

application and on its efficient performance. A lot of effort is been done to improve energy storage capabilities, in addition to implementing nodes that can harvest energy from the environment.

- Scalability: It is the ability to maintain the performance characteristics regardless the size of the networks. As the number of sensor nodes can become very large in some applications, architectures and protocols must provide appropriate support to preserve the services supplied by the network.

- Maintainability: The network have to adapt itself to the changes in the environment and nodes that form it. The cost of replacing such nodes or parameters of the network is usually high.

- Flexible programmability: It is the capability of the sensor nodes to modify the processing options of the acquired data and to perform adjustments in their tasks.

In order to achieve the requirements of the different applications, new mechanisms for communications, architecture and protocol aspects need to be development. The challenge is to identify the ones that work best for a given application. The most common mechanisms considered in WSNs are:

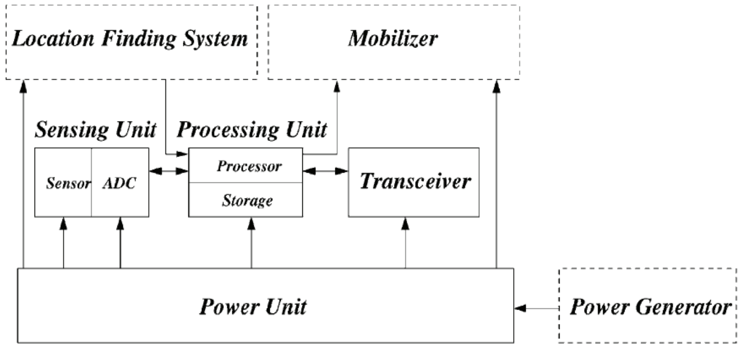


Figure 1.3: Basic components of a Sensor Node

- Multihop wireless communication: Direct communication between nodes and sinks is not always possible. Therefore, using intermediate nodes to be able to send data over long distance is elementary. In addition, this can let to a reduction in the energy required to transmit the packets.
- Energy efficient operation: Energy is the major constrain within WSN characteristics. It is key to find mechanisms that offer and support long time operation to the network.
- Self-configuration: Sensor nodes should be able to adapt its operation parameters, to deal with other node failures, obstacles and changes in the environment.
- Collaboration between nodes: Some applications required a group of sensor nodes to interact in order to detect an event or to make a more complete processing of the information. Some of the collaboration approaches are data aggregation, which reduce the amount of data transmitted while improving the energy efficiency of the network, or distributed processing of the information.
- Data-centric: It is common in many real networks to deploy sensor nodes in a redundant way, to protect the network against failures. In a data-centric view, the identity of the particular node that provides data becomes irrelevant.

1.1.4 Hardware of Wireless Sensor Nodes

Basically, as seen in figure 1.3, a sensor node consists of four components: a sensing unit, a processing unit, a transceiver unit and a power unit. Other components can be added to the node if the desired application requires it: location finding system (e.g. GPS, if the knowledge of location with accuracy are required), a power generator (an/many energy harvesting board/s that support the power unit) and a mobilizer (when node's movement is necessary to carry out the assigned tasks).

Sensing units are usually composed of two subunits: sensors and analog to digital converters (ADCs). Data acquired by sensors need to be converted to digital signals by the ADC, before being sent to the

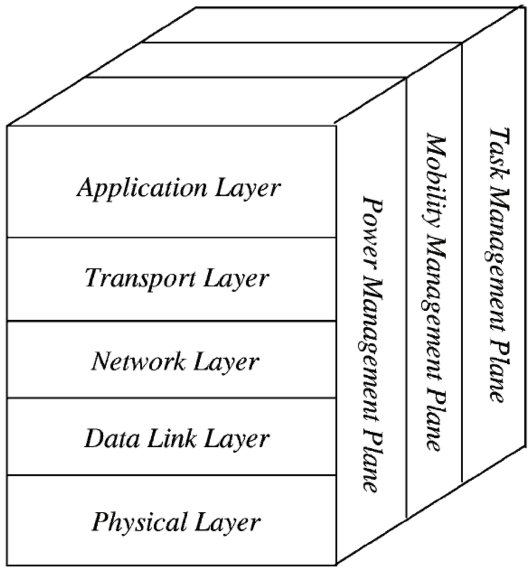


Figure 1.4: The sensor networks protocol stack

processing unit. The processing unit, which is generally features a small storage unit to store the data, manages the tasks analyzing the information that the sensors give and collaborating with the other nodes. The transceiver unit is the gate to the outside through of which the node communicates with the other nodes and the base stations. Finally, the power unit takes care of supplying energy to the system.

The most important constraints that sensor nodes' designers need to address are: the small size, extremely low power consumption, high volumetric densities operation, low production costs necessity, ability to operate unattended and adapt to the environment. Lifetime limitations are probably the more critical, because sensor nodes may need to be deployed in some inaccessible places.

1.1.5 Sensor Networks communication architecture

Sensor nodes have the ability to collect data and route other's data back to the sink and the end users. Nodes use a multihop infrastructureless architecture to route the data back to the sink, which may communicate with the task manager node and the end user via Internet or Satellite. Nodes and sinks use a protocol stack that combines power management, routing awareness, networking protocols and cooperative communication. Figure 1.4 gives an overview of this protocol stack. It consists of the application layer, transport layer, network layer, data link layer, physical layer, power management plane, mobility management plane, and task management plane.

Application layer depends on the task of the nodes and different software can be built in this layer to perform it. The transport layer helps to maintain the flow of data when the sensor network application requires it. The network layer takes care of routing the data supplied by the transport layer. The MAC protocol must account for node's movement, if they are mobile, be power aware and able to minimize collision with neighbors broadcast. The physical layer addresses the needs of a simple but robust modulation, transmission and receiving techniques. In addition, the power, mobility, and task management planes monitor the power, movement, and task distribution among the sensor nodes. These planes help the sensor nodes coordinate the sensing task and lower the overall power consumption.

The power management plane manages the way a node uses its energy. The sensor node may turn off its receiver after receiving a message from one of its neighbors, to avoid getting duplicated messages. Additionally, the node may broadcast to its neighbors that it has low energy and will not be able to participate in routing messages, because the remaining power is used for sensing. The mobility management plane detects the movement of sensor nodes, in order to maintain always a route to the sink, keeping the nodes inside the network. Movement information can also be send to the neighbors, which need to be well known so sensor nodes can balance their power and task usage. The task management plane balances and schedules the sensing tasks given a specific application and the area where the node operates. Not all sensor nodes in that region are required to perform the sensing task at the same time. These management planes are required, so that sensor nodes can work together in an energy efficient way, while routing data in a mobile sensor network, and sharing resources between sensor nodes. Without this collaboration, each sensor node will just work individually, and they will not take advantage of their resources, reducing the network's lifetime.

1.2 Energy Harvesting

1.2.1 Introduction

Energy efficiency is one of the major concerns for the scientific community in the last decades. Therefore, having an additional source of power will increase the lifetime and possibilities of whatever system. Ongoing power management developments enable electronic devices to operate longer for a given power supply, like a battery or a supercapacitor. However, this supplies have their limitations and energy harvesting/scavenging is a complementary approach that has received a lot of attention. Energy harvesting is a way to power wireless sensor networks by scavenging many low grade ambient energy sources such as environmental vibrations, human power, thermal sources, solar, wind and convert into usable electrical energy. Energy harvesting has gained importance because replacing batteries in such sensors is usually expensive. The goal is to achieve power sources that operate over a wide range of environmental conditions and extend nodes lifetime with high reliability.

Although new materials are improving various battery components, their energy capacity improvements are slow. As a result, the minimum required current consumption of sensor nodes often exceeds the rated current capacity of most battery types, leading to suboptimal battery lifetime. Since batteries wear out with time, regular replacement is an inevitable part of maintenance, becoming a major time and resources-consuming task. In many applications do not allow battery changes. For example, re-

Energy Source	Characteristics
Solar	Ambient, Uncontrolable, Predictable
RF Energy	Ambient, Partially controllable
Body heat, Breathing and Blood Pressure	Pasive power, uncontrollable, unpredictable
Finger motion and Footfalls	Active power, Fully controlable
Vibrations	Ambient, Uncontrolable, Unpredictable

Table 1.1: Characteristics of energy sources

placements are not practical in difficult access areas, biomedical implants and in structure-embedded micro-sensors. Moreover, the increasing number of battery-powered sensors is driving a disposal problem with important environmental consequences.

There are other advantages of using energy harvesting, including the ability to monitor more closely the amount of energy being used by a system and hence have an improved level of energy-awareness, which may be required for state-of-the-art sensor network management algorithms.

1.2.2 Sources of Energy

Energy harvesters usually consist of three main components: the microgenerator that converts ambient energy into electrical energy, the voltage booster which regulates the generated voltage, and the storage element which can be a supercapacitor or a battery. The main sources of available energy can be classified into the following categories: mechanical, light, thermal, electromagnetic, and human. Table 1.1 lists the characteristics of some of the energy sources cited.

Mechanical Energy: Some devices can produce energy from vibration, mechanical stress and strain of the surface the sensor is deployed on. Energy from vibrations is generally based on the movement of a spring-mounted mass relative to its support frame. Those vibrations produce mechanical acceleration that makes the mass component move and oscillate. This energy can be converted into electrical energy via a magnetic field (electromagnetic), strain on a piezoelectric material or an electric field (electrostatic). Most of these systems rely on resonance to work, implying that there is a peak frequency at which the system obtains most part of the energy. Therefore, the stress level in the layers deposited on the oscillating structure is an important parameter for maximization of the power output of a vibration energy harvester. Some devices can work in a slightly different frequency when there is an excess of energy available. In addition to vibrations, mechanical energy harvesting can rely on natural sources such as wind and water flow.

Light Energy: Photovoltaic cells can produce energy out the ambient light either in indoors or outdoors scenarios. Solar energy is the most powerful source of light, but other sources can be used in the indoors case, like incandescent, fluorescent and diode. Photovoltaic cells convert light energy into electricity at the atomic level. When light strikes the cell, a certain portion of it is absorbed within a semiconductor material, where it knocks electrons loose, allowing them to flow freely. Silicon is currently the most common because of its conversion efficiency. The major challenge is to conform to small surface area and its power output strongly depends on environmental conditions.

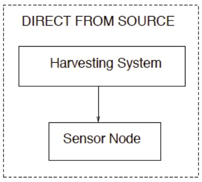
Thermal Energy: Thermoelectric energy harvests exploit the Seebeck effect, according to which electricity is generated from a temperature difference between opposite segment of a conducting material. Such temperature differences result in heat flow and, as a result, charge flow. Generally, thermoelectric devices require a large and sustained temperature gradient between two surfaces in order to provide useful power. In microsystem is difficult to observe differences greater than 10C. Thermoelectric power generation presents many advantages including solid-state operation with no moving parts, long life, and high reliability. However, they have a large size and have low efficiency.

Electromagnetic Energy: A large number of potential RF sources, such as broadcast radio and TV, mobile telephony or wireless networks, are present in high populated areas. It is possible to collect part of the energy produced by these sources and convert it into useful energy. The conversion is based on a special kind of rectifying antenna that can convert microwave energy into DC electricity. This type of antenna is called rectenna, and they are highly efficient at converting microwave energy to electricity. However, the energy levels actually present are low, which makes difficult for electronic devices to use it apart from the cases when they are close to the real source of RF energy.

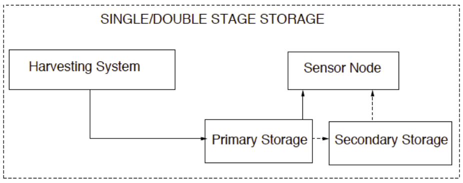
Energy from the Human Body: The human body is continuously moving and radiating heat. Even when resting, it can emit about 100W into the environment. It is possible to tap into some of this energy to power wearable electronics that can form a Body Sensor Network. We may distinguish between active, where users are required to perform a specific task, and passive energy harvesting methods, that harvest energy from the users everyday actions, e.g. walking, breathing, body heat, blood pressure...

1.2.3 Energy Harvesting in WSN

Previously, sensor networks were designed with finite energy as the primary constraint. A sensor network may be optimized to increase its lifetime by operating nodes at low duty-cycles while compromising sensing reliability in the process; whereas a network optimized for reliability and coverage will have to operate either with larger batteries, or involving periodic cost to change batteries, or a redundant deployment. As a result, nodes powered only by batteries do not meet the requirements of these potentially conflicting application design goals; [14]. Thanks to energy harvesting and recharge opportunities, the constraint of finite energy is less stringent. Recharge and harvesting opportunities impact both individual node operations as well as system design considerations, which will be discussed in the rest of this subsection.



(a) Harvest-Use



(b) Harvest-Store-Use

Figure 1.5: Energy harvesting architectures

Energy Harvesting Architectures

Energy harvesting techniques can meet partially or totally the energy needs of a sensor node, supporting other sources of power like batteries or capacitors. Broadly, there two main architectures to use energy harvesting capabilities in a sensor node: Harvest-Use and Harvest-Store-Use. Figure 1.5 shows both architectures.

A *Harvest-Use* system directly powers the sensor node and, to assure node operation, the power output of the harvesting system has to be continuously above the node's minimum voltage. If the energy available is not enough, the node will be disabled, making the sensor node to oscillate in on and off states. These kind of architecture can be used to harvest from mechanical energy like pushing buttons, walking, pedaling, etc.

The *Harvest-Store-Use* architecture consists of a storage component that stores the harvested energy and powers the sensor node. Storage is useful when the harvested energy available is more than the current usage. For example, the node may be sleeping while harvesting energy, and use this energy later to send or receive data. In addition, the source the energy harvesting board is using may not be available during some time. Therefore, this architecture use uncontrolled but predictable energy sources like solar energy.

Storage technologies

Part of the efficiency of the energy harvesting boards depends on the store technology used. The choice of the storage component and the corresponding recharge technology is of prime significance for energy harvesting systems. Rechargeable batteries, a common choice of energy storage, are made up of several technologies (chemical compositions). A rechargeable battery is a storage cell that can be charged by reversing the internal chemical reaction. A few of the popular rechargeable technologies are Sealed Lead Acid (SLA), Nickel Cadmium (NiCd), Nickel Metal Hydride (NiMH) and Lithium Ion (Li-ion). These battery technologies can be characterized along several axes - energy density, power, storage efficiency, discharge rate and number of deep recharge cycles - which typical values can be observed in Table 1.2.

Technology	**Energy Density** (MJ/kg)	**Power Density** (W/kg)	**Efficiency** (%)	**Discharge Rate** (% per month)	**Recharge Cycles**
Sealed Lead Acid	0.11-0.14	180	70-92	3-4	500-800
Ni-cadmium	0.14-0.22	150	70-90	20	1500
NiMH	0.11-0.29	250-1000	66	20	1000
Li-ion	0.58	1800	99.9	5-10	1200

Table 1.2: Rechargeable battery technologies

Li-ion storage technology is envisaged to be the best for storing harvested energy. These batteries have high output voltage, energy density, efficiency and moderately low self-discharge rates. Alternatively, super-capacitors can be used instead of or along with rechargeable batteries as storage components. Although their self-discharge rate is higher than batteries, super-capacitors have theoretically infinite recharge cycles.

Node-level Adaptations

Each node can independently, or in coordination with other nodes, estimate its residual energy and harvesting possibilities to tune parameter settings, positively influencing performance metrics. Followingis a list of potential system parameters that a node can tune based on its effective energy estimate.

Duty-Cycling: The duty-cycle is the fraction of time a node is ON in a cycle of ON and OFF durations. With a higher duty-cycle, the node uses energy at a quicker rate, which provides higher sampling reliability and lower communication delay. However, this increases the energy consumption. Traditionally, variations in the duty-cycle are done to meet a lifetime requirement based on the finite energy constraint. With energy-harvesting opportunities, the energy constraint is relaxed and the duty-cycle parameter can be tuned more often and possibly maintained at higher levels depending on the prediction of available energy.

Transmitting power: The wireless radio is one of the major consuming components within a sensor node. Moderating transmit power is one of the mechanisms to reduce the energy used for communication. Transmitting power can be adapted to increase the communication range of a node and increase efficiency of data dissemination protocols.

Sensing Reliability: Sensing reliability is the quality of information provided by the sensor. Similar to a nodes duty-cycle and transmit power, sensing reliability has a tradeoff with energy usage. One of the parameters that can be tuned is the nodes sensing frequency which increases the sensing reliability. In energy harvesting nodes, sensing reliability can be varied proportionally with the predicted effective energy. In addition, the amount of data processed at a sensor node can be varied proportionally with the predicted effective energy to obtain the highest sensing reliability possible.

Transmission Scheduling: A way ti reduce the number of transmissions that a node makes is to aggregate data from various nodes at intermediate nodes and transmit fewer messages. Similarly, data at a single node can be aggregated in temporal manner to send collective information and reduce the number of transmissions. Energy harvesting nodes can be responsible for data aggregation and transmission. Further, these nodes can schedule their data dissemination based on predicted effective energy.

Implications on System Design

We have seen how individual nodes can exploit the presence of a harvestable energy source and improve performance parameters. Energy harvesting nodes' presence within a network has an impact on the network-wide solutions and protocols. Some of the changes to traditional solutions the network use to take advantage of energy harvesting are:

Network Deployment: Traditionally, sensors were deployed in a redundant and dense manner to increase lifetime. With this redundancy, nodes coordinate and operate in smaller groups to cover and sense the entire region of interest. Using energy harvesting nodes helps to decrease the total number of nodes. Predicting the availability of the desirable energy used to harvest such nodes, as well as the rate of energy usage, less nodes need be placed in the area to maintain a similar sensing coverage. Consequently, this reduces to cost of the network deployment.

Routing: Sensor routing protocol incorporate energy-awareness techniques for routing decisions. The expected harvestable energy can be also taken into account to improve these decisions. Usually, the best route is the one that delivers the first response from the source to a data request query from sink. Other route selection metrics include delivery probability and number of hops. Considering a network consisting of battery-powered and energy harvesting sensor nodes, energy harvesting nodes can be selected as intermediaries for routes, while battery-powered nodes will then be used only as last hop nodes as long as it exists an energy harvesting neighbor. An increase of the duty-cycle of each node on a route affects the end to end latency, decreasing the total delay. Although using an effective energy metric may not yield the shortest path, the least latency path or the most reliable path, maximizing the number of energy harvestable nodes on routing paths has the potential to increase the overall lifetime of the network.

Clustering: Another useful mechanism to route packets in sensor networks is through formation of clusters. Nodes route packets to cluster heads, which in turn, transmit packets on a cluster-head overlay to reach the destination. Harvestable nodes, with regular recharge opportunities, are suitable candidates to be chosen as cluster heads and battery-powered nodes may route their packets over a few hops to cluster heads and consume less power as compared to them.

Chapter 2

MAKERS: Multiple Board Markov Model for Energy Harvesting Sensors

2.1 Introduction

The goal of EH networks is to ensure continuous network operation, wherein nodes may regulate their transmission and harvesting cycles judiciously to meet network constraints. Specifically, in this paper, we assume a network architecture composed of wireless sensor networks (WSNs), though the same concepts can be trivially extended for other types of networks. Only recently has the research community engaged in developing higher layer network protocols for EH networks [7] [6] [10], though the underlying principles of harvesting energy from vibrations, solar, and wind are well known. Despite these strides in the area of protocol design for EH networks, the development of theoretical models for energy harvesting, and the prediction of the residual energy state during an ongoing network operation are still in a nascent stage. Analytical formulations for the case of a single harvesting source have been previously presented in [11] [21], which we extend significantly in this paper as follows:

- We develop a Markov model for capturing the energy states of the sensors equipped with multiple energy harvesting boards. These sensors can harvest energy from the same source, or a combination of different sources (such as vibration and solar). This presents a general case for EH WSN design, as a single source cannot be assumed to be always present during network operation.

- We provide simplified analytical models for predicting the probability of a sensor running out of energy (hence, mis-detecting the event, which we call as the *event-loss*), and the time before this occurs. Compared to earlier work [11] [21], not only are our models lower in complexity facilitating on-board computation in the sensors, but can also be applied for sensors with multiple harvesting boards.

The rest of this chapter is described as follows: In Section 2.2, we list the general assumptions taken in the model. In Section 2.3 we develop the MAKERS model. We derive analytically the *event-loss* probability and average time before this occurs in Sections 2.4 and 2.5, respectively. We take a look to the performance evaluation in Section 2.6. Finally, in Section 2.7, we conclude our work.

2.2 Preliminary Discussion and Assumptions

The MAKERS model assumes the case of multiple boards, capable of harvesting energy from different sources. Before we present the general model, we develop our analysis under the constraint that all the M EH boards connected to each sensor harvest energy from the same source a (say, all are solar panels). Later, we relax this constraint to account for multiple sources. Note that many of the notations used in this section are similar to the earlier work in [11].

Each EH board (out of the total M) can be independently in the *active* (i.e., currently harvesting with a rate ρ_a) or in the *inactive* (off) state. Further, the time duration for which the sensor stays in these two states are exponentially distributed with the means T_{on} and T_{off}, respectively. The probability of changing from *active* to *inactive* is r, and the reverse, is w. Hence, the overall probability of the EH board to be *active* is:

$$\mu = \frac{w}{r+w} \tag{2.1}$$

An event that needs to be sensed and reported occurs with a probability equal to p. The time between these sensing events, t_p, is exponentially distributed with mean T_p. Each event consumes a total amount of energy equal to E, which includes the energy expended nodal processing, as well as transmission or reception of the data. If there is no event during a time slot, no energy is consumed (the on-board sensors are passive components), and battery leakage is negligible.

We note that the required energy E for a given sensing event may not be completely harvested within a single time unit T. Thus, multiple slots of duration T may be needed to obtain sufficient energy within the node, which is given as $k = \frac{E}{\rho_a T}$ [21]. Let $i \in [0, M]$ be the number of *active* boards in a time slot. Without loss of generality, we assume that $1 \leq M \leq k$, where M is a positive integer and k a positive real number. In addition, the node incorporates a battery or a super-capacitor, with a storage capacity equal to $(N-1)E$. Hence, a full battery will allow us to run $N-1$ events.

The time unit or slot used in this model is T, where $T \ll T_a, T_i, T_p$. Hence, only one event can occur in a time slot. Therefore, the probabilities r, w and p described above can be found as follows, as derived earlier in [21]:

- From *active* to *inactive*: $r = \frac{T}{T_a} e^{-\frac{T}{T_a}} \approx \frac{T}{T_a}$
- From *inactive* to *active*: $w = \frac{T}{T_i} e^{-\frac{T}{T_i}} \approx \frac{T}{T_i}$

- Event occurrence: $p = \frac{T}{T_p} e^{-\frac{T}{T_p}} \approx \frac{T}{T_p}$

Note that while distributions other than the exponential are possible for the event intervals, the meaning of the variables r, w and p in the overall system model will remain unchanged.

2.3 The MAKERS Model

In this section, we develop our proposed analytical model. Our approach will be to start first with a finer granularity of a $(M+1)kN$ states model, that also splits up each inter-sensing event duration into k sub-states (recall k fractional harvesting durations give the energy required for the sensing event). Then, by merging the sub-states every k, we build the general $(M+1)N$ model. Finally, we describe how $M+1$ merge into 1 and establish a simplified version of the model with N states, which shall help in the following sections.

The MAKERS general model has $(M+1)N$ states, a product of the $M+1$, from 0 to M *active* harvesting boards, and N, which represents the amount of energy left in the battery in the current time slot. During a time slot, the node harvests a total amount of energy equal to $i\rho_a T$ and spends E if an event occurs. Moreover, the energy that is being harvested during the current time slot can not be used to run an event that happens in it, unlike the assumptions made in [11]. In other words, if we have residual energy $\frac{k-1}{k}E$ in the battery, we will not be able to run an event during this time slot even though many of the boards are *active*.

2.3.1 The $(M+1)kN$ states model

We begin with a model with $(M+1)kN$ states that can be enumerate as $s = iN + n$, where s is the state number from 0 to $(M+1)kN - 1$, i is the number of *active* boards and n is the energy left, from 0 to $(N-1)k$, in $\frac{1}{k}$ steps. In addition, the amount of energy stored in the battery is given by $B = s \bmod_N$. Hence, the battery is empty in state 0, N, $2N\ldots$ and totally full in $N-1$, $2N-1\ldots$ Firstly, we focus on how the system changes from one state to another with respect to the battery life. In figure 2.1, we show the Markov chain for the $(M+1)kN$ model in the case of $M=2$ and $k=3$, with the transition probabilities from the states having the same energy n. Each horizontal row of states corresponds to a number of active boards (hence, only three rows are shown as $M=2$), while the vertical columns are energy sub-states, with three columns ($k=3$) between the two residual energy state n and $n+1$. Let $\delta_{i,j}$ be the probability of j harvesters *active* in the future state if i harvesters are *active* in the current one.

For example, if we are in state with residual energy n and 1 *active* board (2*nd* row, 4*th* col.) and an event occurs, the node will consume E at the same time that harvests $\frac{1}{3}E$, so the future state will have a residual energy equal to $n - \frac{2}{3}$. Then, 3 possible transitions may occur depending on the future number of *active* boards, described by $\delta_{1,j}$.

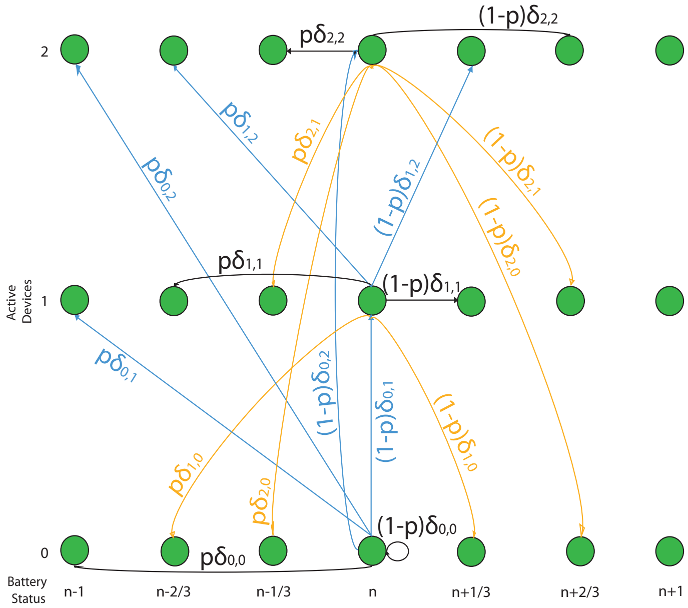


Figure 2.1: $(M+1)kN$ states model, $M=2$, $k=3$ centered at the state n

For the states corresponding to the special cases of the battery completely *full* or *empty*, the transition probabilities are slightly different to the ones presented. This can be observed in figure 2.2, where the transition probabilities from states with energy $\frac{1}{3}$ (states with energy 0 and $\frac{2}{3}$ follow the same transitions) and $N-1$. We have also included the transition probabilities from state $N-1-\frac{1}{3}$ and 2 *active* boards, because they experiment changes compared with the case of an arbitrary state n. Even though the node is harvesting $\frac{2}{3}E$, when an event does not occur, the maximum residual energy the node will have is $N-1$.

2.3.2 The $(M+1)N$ states model

We aim to find a generic form for the transition probability from one state to another. Thus, every k energy levels will be merged into one, to create the simplified $(M+1)N$ model. For example, the first k states in terms of *battery life*, called 0, $\frac{1}{k}$, ..., $\frac{k-1}{k}$; will be merged in a single state representing the residual energy 0. From figure **??** we can formally define the transition probabilities $p_{a,b/i,j}$, where a is the current energy level, b the future energy level, and the number of current and future *active* boards is given by i and j, respectively:

- $p_{n,n-1/i,j} = p\frac{k-i}{k}\delta_{i,j}$
- $p_{n,n/i,j} = [(1-p)\frac{k-i}{k} + p\frac{i}{k}]\delta_{i,j}$
- $p_{n,n+1/i,j} = (1-p)\frac{i}{k}\delta_{i,j}$

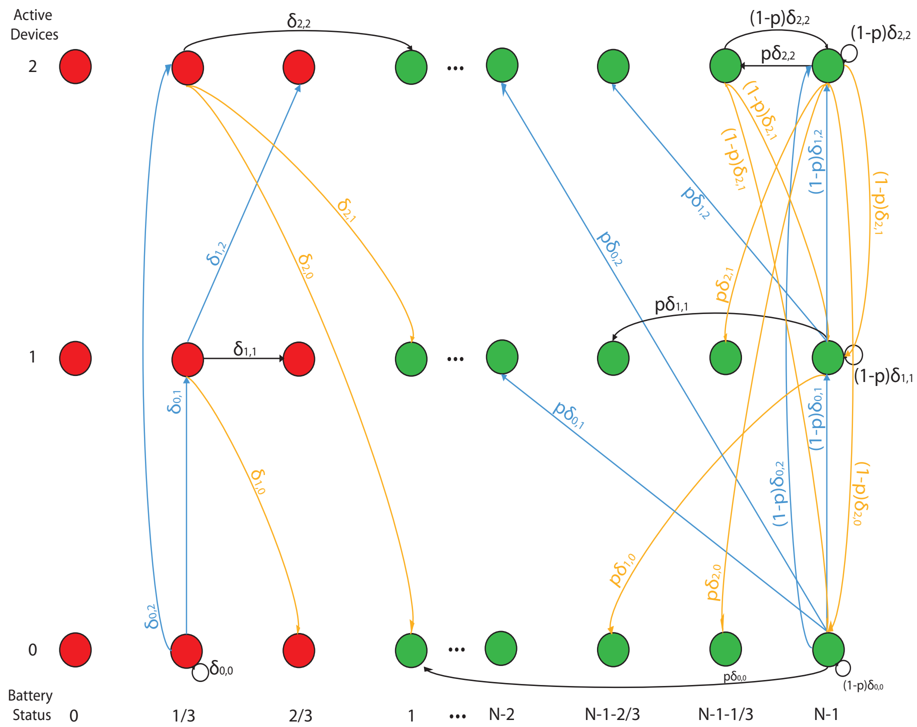


Figure 2.2: $(M+1)kN$ states model, $M=2$, $k=3$ for states 1/3 and N-1

Note that from an arbitrary state n we can only reach another state with a battery level that differs by a single unit, i.e., only states with battery level equal to $n-1$, n and $n+1$ are possible, with other transition probabilities set to 0. Moreover, if an event happens in state n we will not be able to reach the state $n+1$ (higher energy). Similarly, $n-1$ (lower energy) can only be reached if an event occurs. In figure 2.3 we can see the Markov chain for the $(M+1)N$ model for $M=2$ and a generic k with the transition probabilities from the states with energy n represented.

Again, the two boundary states with energy level 0 and $N-1$ will have different probabilities for some cases. These special cases can be observed in 2.4 and they are expressed as follows:

- $p_{0,0/i,j} = \frac{k-i}{k}\delta_{i,j}$
- $p_{0,1/i,j} = \frac{i}{k}\delta_{i,j}$
- $p_{N-1,N-1/i,j} = [(1-p)+p\frac{i}{k}]\delta_{i,j}$

In order to obtain $\delta_{i,j}$, i.e., the probability of j harvesters being *active* from an earlier total number i, we take into account all the possible combinations of *active*-*inactive* changes that may happen. We observe that there are two possibilities:

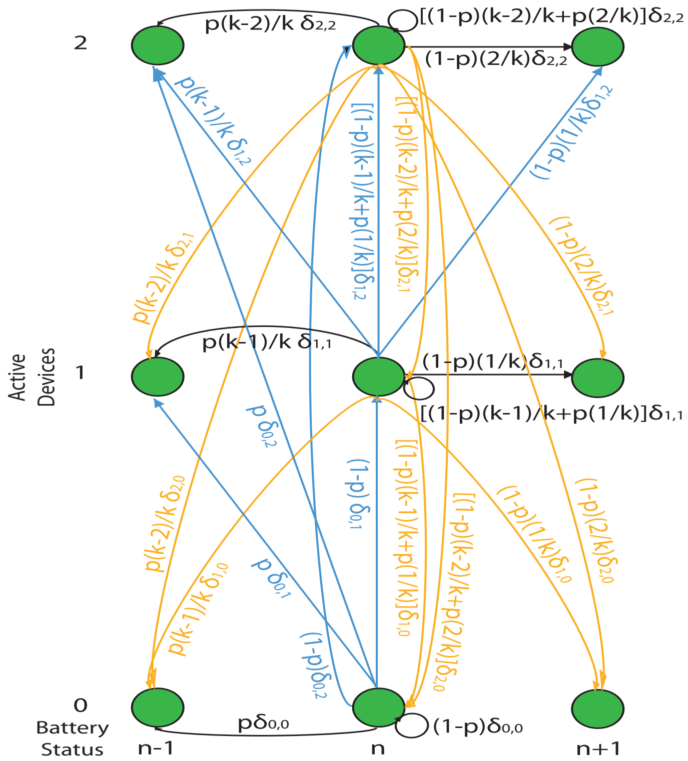


Figure 2.3: (M+1)N states model, M=2, generic k focused in n

First, when $j \leq i$, $i-j$ boards need to change their state from *active* to *inactive* while the rest remain in the same state. However, many other possibilities can occur because some other currently *active* boards can switch to *inactive* if the same number of *inactive* boards switch as well. Similarly, when j is bigger than i, we have to take into account the different combinations, all of which have finite occurrence probability.

In table 2.1, $\delta_{i,j}$ in all cases for $M=4$ are derived. We are going to focus on two specific cases to guide the reader through the table:

For $i=2$ and $j=2$, the first term, $(1-r)^2(1-w)^2$, represents the case when the two currently *active* boards do not switch their state and the two *inactive* neither. The second term, $4(1-r)r(1-w)w$, represents the case when one of the *active* boards switches to *inactive*, forcing one of the *active* to switch in order to get $j=2$. Here, 4 represents the number of combinations that this can happen: $\binom{2}{1}\binom{2}{1}=4$, meaning that whichever of the two *active* may switch and then whichever of the *inactive* can change. Finally, the third term, r^2w^2, represents the case where both *active* boards switch forcing the two *inactive* to do so.

For $i=1$ and $j=3$, the first term, $3(1-r)(1-w)w^2$, represents the case when the currently *active* board do not switch its state and the two *inactive* change it to *active*. 3 represents the number of possible combinations: $3=\binom{1}{1}\binom{3}{2}$. The second term, rw^3, represents the case when the *active* board switches to

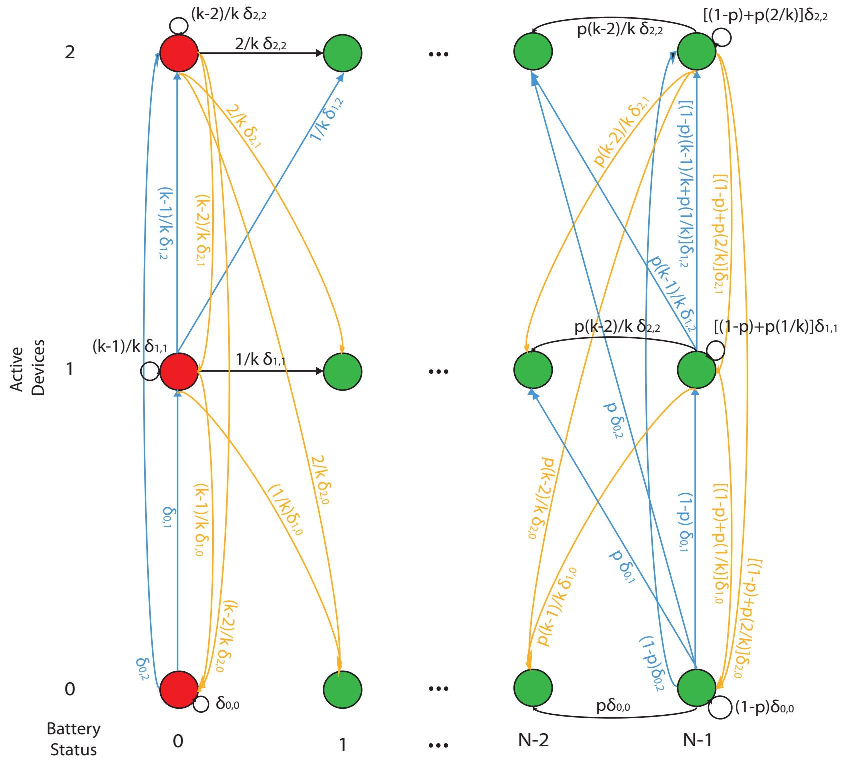


Figure 2.4: (M+1)N states model, M=2, generic k focused in 0 and N-1

inactive and all the *inactive* become *active*.

Note that $\sum_{j=0}^{M}\delta_{i,j}=1$, $\forall i$, as expected. Remembering that $\binom{m}{0}$ and $\binom{m}{m}$ are always equal to 1, considering m a natural number, we can state the expressions for $\delta_{i,j}$ for the above two cases $i<j$ and $j\leq i$, which are obtained through straightforward but tedious analytical derivations:

$$\delta_{i,j}=\begin{cases}\sum\limits_{l=0}^{\min(M-i,j)}\binom{i}{j-l}\binom{M-i}{l}(1-r)^{j-l}r^{i-(j-l)}(1-w)^{M-i-l}w^{l} & 0\leq j\leq i\\ \sum\limits_{l=0}^{\min(M-j,i)}\binom{i}{l}\binom{M-i}{M-j-l}(1-r)^{i-l}r^{l}(1-w)^{M-j-l}w^{j-(i-l)} & i<j\leq M\end{cases} \tag{2.2}$$

Once these state transition probabilities are calculated, the MAKERS model is completely represented. Next, we describe a simplified version of this model, which shall help in obtaining closed form equations about the performance of the device.

2.3.3 Simplified N states model

In this model, the state only denote the battery status, from 0 to $N-1$, and the states of the individual boards do not appear in the state definition. In every energy level, the case of having any arbitrary number of *active* boards is considered. Recall that the probability of a board to be *active*, given by equation 2.1,

		j				
		4	3	2	1	0
i	4	$(1-r)^4$	$4(1-r)^3r$	$6(1-r)^2r^2$	$4(1-r)r^3$	r^4
	3	$(1-r)^3w$	$(1-r)^3(1-w)+3(1-r)^2rw$	$3(1-r)^2r(1-w)+3(1-r)r^2w$	$3(1-r)r^2(1-w)+r^3w$	$r^3(1-w)$
	2	$(1-r)^2w^2$	$2(1-r)^2(1-w)w+2(1-r)rw^2$	$(1-r)^2(1-w)^2+4(1-r)r(1-w)w+r^2w^2$	$4(1-r)r(1-w)^2+2r^2(1-w)w$	$r^2(1-w)^2$
	1	$(1-r)w^3$	$3(1-r)(1-w)w^2+rw^3$	$3(1-r)(1-w)^2w+3r(1-w)w^2$	$(1-r)(1-w)^3+3r(1-w)^2w$	$r(1-w)^3$
	0	w^4	$4(1-w)w^3$	$6(1-w)^2w^2$	$4(1-w)^3w$	$(1-w)^4$

Table 2.1: $\delta_{i,j}$ in the case of $M = 4$

the probability of having i *active* boards, ϕ_i, is:

$$\phi_i = \binom{M}{i}\mu^i(1-\mu)^{M-i} \tag{2.3}$$

The sum of transition probabilities from one energy state to another, or to the same state (states defined in terms of energy level alone) remain the same as the general MAKERS model in Section 2.3.2. Hence, we only take into account the number of *active* boards in the current state, which leads us to a very simple model. We define α, a variable that we use subsequently, as follows:

$$\alpha = \sum_{i=0}^{M}\frac{i}{k}\phi_i \tag{2.4}$$

Note that α is the weighted sum of the energy harvested for the i *active* boards multiplied per the probability of having i *active* while the rest, $M-i$, remain *inactive*.

Now, the transition probabilities for this simplified model is represented as $p_{a,b}$, where a is the current energy level, b the future one. In addition, from the figure 2.5, we can express $p_{a,b}$ as follows:

- $p_{n,n-1} = p(1-\alpha)$
- $p_{n,n} = (1-p)(1-\alpha)+p\alpha$
- $p_{n,n+1} = (1-p)\alpha$

In the boundaries states, the transition probabilities are slightly different from the ones stated previously. They are represented in figure 2.5 and derived as:

- $p_{0,0} = p(1-\alpha)$

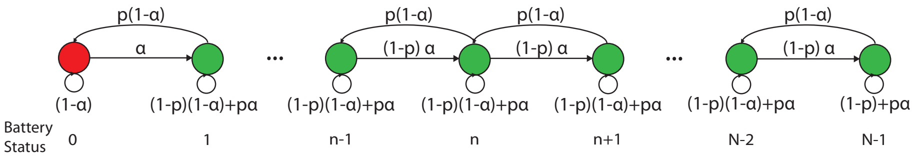


Figure 2.5: Simplified N states model

- $p_{n,n} = (1-p)(1-\alpha) + p\alpha$
- $p_{n,n+1} = (1-p)\alpha$

Using this simplified expression, we can now compute the closed form expressions for the probability of *event-loss*, and the average time to its occurrence.

2.4 Probability of Event-Loss

An event is lost (i.e. is not sensed and reported), if a node lacks sufficient energy E to process it. This probability, P_L, can be found solving the equation system defined by $\mathbf{P}\pi = \mathbf{P}$ and $\sum_{n=0}^{N-1} \pi(n) = 1$, where $\mathbf{P}$ is the transition probability matrix, π is the vector of the steady probabilities for each state, from 0 to $N-1$. Therefore, the loss probability will be given by $P_L = \pi(0)$. The simplified Markov chain model allows us to derive a closed form solution. Towards this aim, we analyze the equilibrium equations for different states. We begin from the state 0, which can be written as:

$$\pi_0\alpha = \pi_1 p(1-\alpha) \tag{2.5}$$

We continue with state 1:

$$\pi_1(p(1-\alpha)+(1-p)\alpha) = \pi_0\alpha + \pi_2 p(1-\alpha) \tag{2.6}$$

If we replace π_0 using equation 2.5, we find:

$$\pi_1 = \frac{p(1-\alpha)}{(1-p)\alpha}\pi_2 \tag{2.7}$$

Hence, for an arbitrary state n, $1 < n < N-1$:

$$\pi_n(p(1-\alpha)+(1-p)\alpha) = \pi_{n-1}(1-p)\alpha + \pi_{n+1}p(1-\alpha) \tag{2.8}$$

This time, we analyze it in the case of $n = 2$:

$$\pi_2(p(1-\alpha)+(1-p)\alpha) = \pi_1(1-p)\alpha + \pi_3 p(1-\alpha) \tag{2.9}$$

Now, we use equation 2.8 to substitute π_1, therefore:

$$\pi_2 = \frac{p(1-\alpha)}{(1-p)\alpha}\pi_3 \tag{2.10}$$

Notice that expression for π_2 is similar to 2.7, for π_1. Continuing the analysis for state $N-1$:

$$\pi_{N-1}p(1-\alpha) = \pi_{N-2}(1-p)\alpha \tag{2.11}$$

Observing the similar pattern as seen in earlier equations 2.7, 2.10 and 2.11, we express π_{n+1}, $1 \leq n < N-1$, as:

$$\pi_{n+1} = \frac{(1-p)\alpha}{p(1-\alpha)}\pi_n \tag{2.12}$$

We define γ as follows, to simplify future equations:

$$\gamma = \frac{(1-p)\alpha}{p(1-\alpha)} \tag{2.13}$$

Combing the equation 2.12 and the one for state 0, eq. 2.5, we rewrite π_n, $1 \leq n \leq N-1$, as:

$$\pi_n = \frac{\gamma^n}{(1-p)}\pi_0 \tag{2.14}$$

The last step is to find π_0, or P_L. We know that the sum of all the steady state probabilities must be equal to 1. Therefore, we can re-write 2.5 as follows:

$$\begin{gathered}\pi_0(1+\textstyle\sum_{n=1}^{N-1}\frac{\gamma^n}{(1-p)}) = 1 \\ \pi_0(1+\frac{(1-\gamma)}{(1-p)(1-\gamma)}\textstyle\sum_{n=0}^{N-1}\gamma^n - \frac{1}{(1-p)}) = 1\end{gathered} \tag{2.15}$$

We could insert the following form for the series in the previous equation 2.15, with the constrain $0 < \gamma < 1$, so $0 < (1-p)\alpha < p(1-\alpha)$.

$$(1-\gamma)\sum_{n=0}^{N}\gamma^n = 1-\gamma^{N+1} \tag{2.16}$$

Finally, we will have two cases for P_L depending on the constraint stated in the last equation:

$$P_L = \begin{cases} \frac{(1-p)(1-\gamma)}{1-\gamma^N-p(1-\gamma)} & (1-p)\alpha < p(1-\alpha) \\ \frac{1}{1+\frac{1}{1-p}\sum_{n=1}^{N-1}\gamma^n} & (1-p)\alpha > p(1-\alpha) \end{cases} \tag{2.17}$$

Returning to the $(M+1)N$ model with its steady state probabilities, $\Pi_{n,i}$, where n is the energy level and i the number of active boards for $0 \leq n \leq N-1$ and $0 \leq i \leq M$, we compute:

$$\Pi_{n,i} = \phi_i\pi_n \tag{2.18}$$

2.5 Average Time before Event-Loss

The analysis of the average time before *event-loss* is calculated using the simplified model in Section 2.3.3 because it allows a straightforward calculation. Let $T_{n\to l}$ be the average number of steps required to visit state l for the first time, starting at state n. With this notation, the average time before *event-loss* for state n is represented by $T_{n\to 0}$. We take advantage of the chair structure of the MAKERS model to recursively calculate $T_{n\to 0}$. Given a start state n, where $0 < n \leq N-1$, we visit state $n-1$ before absorption, running out of energy. Hence, we have:

$$T_{n\to 0} = T_{n\to n-1} + T_{n-1\to 0} \tag{2.19}$$

Thus, if we know $T_{n\to n-1}$, for $1 \leq n \leq N-1$ we can calculate $T_{n\to 0}$ as follows:

$$T_{n\to 0} = \sum_{y=1}^{n} T_{y\to y-1} \tag{2.20}$$

Considering states n and $n-1$, where $1 \leq n \leq N-1$, for $T_{n\to n-1}$ we can write:

$$\begin{aligned}
T_{n\to n-1} &= \sum_{g=0}^{\infty}(g+1)[(1-p)(1-\alpha)+p\alpha]^g p(1-\alpha) \\
&+ \sum_{g=0}^{\infty}(g+1+T_{n+1\to n-1})[(1-p)(1-\alpha)+p\alpha]^g(1-p)\alpha \\
&= \frac{p(1-\alpha)}{(1-[(1-p)(1-\alpha)+p\alpha])^2} + \frac{(1-p)\alpha}{(1-[(1-p)(1-\alpha)+p\alpha])^2} \\
&+ \frac{(1-p)\alpha}{1-[(1-p)(1-\alpha)+p\alpha]}T_{n+1\to n-1} \\
&= \frac{1}{p+\alpha-2p\alpha} + \frac{(1-p)\alpha}{p+\alpha-2p\alpha}T_{n+1\to n-1}
\end{aligned} \tag{2.21}$$

For $T_{n+1\to n-1}$, where $0 < n < N-1$, we have:

$$T_{n+1\to n-1} = T_{n+1\to n} + T_{n\to n-1} \tag{2.22}$$

Applying 2.21 we get:

$$T_{n+1\to n-1} = \frac{1}{p(1-\alpha)} + \frac{p+\alpha-2p\alpha}{p(1-\alpha)}T_{n+1\to n} \tag{2.23}$$

Now, using 2.23 in 2.21, we have:

$$T_{n\to n-1} = \frac{1}{p(1-\alpha)} + \gamma T_{n+1\to n} \tag{2.24}$$

Then, the case of $T_{N-1\to N-2}$, can be found as follows:

$$\begin{aligned}
T_{N-1\to N-2} &= \sum_{g=0}^{\infty}(g+1)[(1-p)+p\alpha]^g p(1-\alpha) \\
&= \frac{p(1-\alpha)}{(1-[(1-p)+p\alpha])^2} \\
&= \frac{1}{p(1-\alpha)}
\end{aligned} \tag{2.25}$$

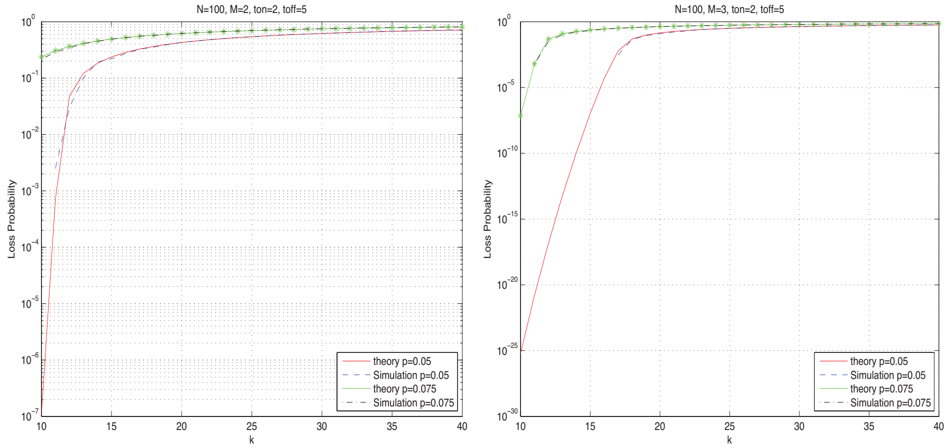


Figure 2.6: Loss Probability (k) for ton=5, toff=5

By recursively applying 2.24 and using 2.25, we get:

$$T_{n\to n-1} = \frac{1}{p(1-\alpha)} \sum_{x=0}^{N-1-n} \gamma^x \tag{2.26}$$

Therefore, using 2.20 and 2.26, we find the Average Time before Event-Loss:

$$T_{n\to 0} = \frac{1}{p(1-\alpha)} \sum_{y=1}^{n} \sum_{x=0}^{N-1-y} \gamma^x \tag{2.27}$$

2.6 Results

Monte-Carlo continuous-time simulations are undertaken in MATLAB to evaluate our approach. The values for t_a, t_i and t_p are randomly generated through exponential distributions with means T_a, T_i and T_p, respectively. The values of these parameters are mentioned in the individual figures. We took a time slot equal to 0.1 seconds, $T_s = 0.1$, and we simulated up to one million times units.

Figures 2.6, 2.7 and 2.8 compare the *event-loss* probability obtained from theory, Eq. 2.17, with those obtained from the simulations. In these figures, we compare the results by varying k. Every harvesting board has its own parameters following exponential distributions, but they have all the same mean time for the state durations, as we assume they have the same features. We have assumed different situations, firstly being T_{on} smaller than T_{off}, the second one taking both the same value and finally for a T_{on} greater than T_{off}. In each figure, we show the cases for $M = 2$ and $M = 3$.

Notice that, as expected, a bigger M makes a lot of difference when k it is not big. That is because when harvesting energy, each board harvest $\frac{1}{k}E$ every time slot, the smaller k is the bigger the percentage of E harvested is.

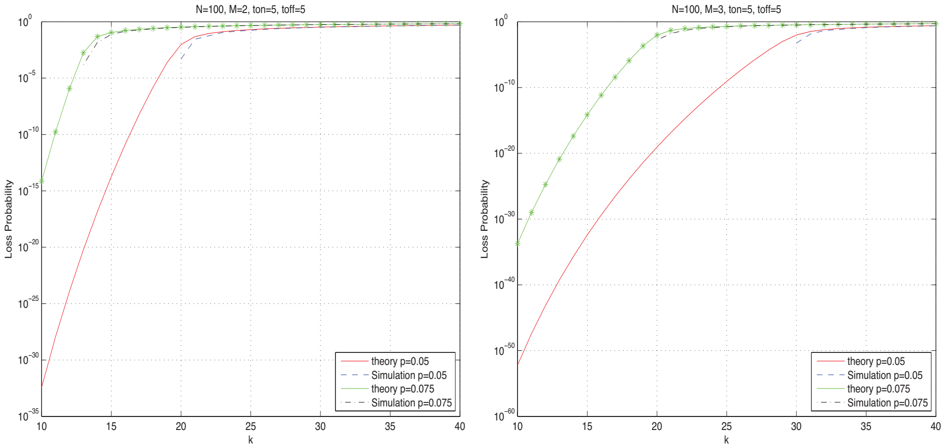


Figure 2.7: Loss Probability (k) for ton=5, toff=5

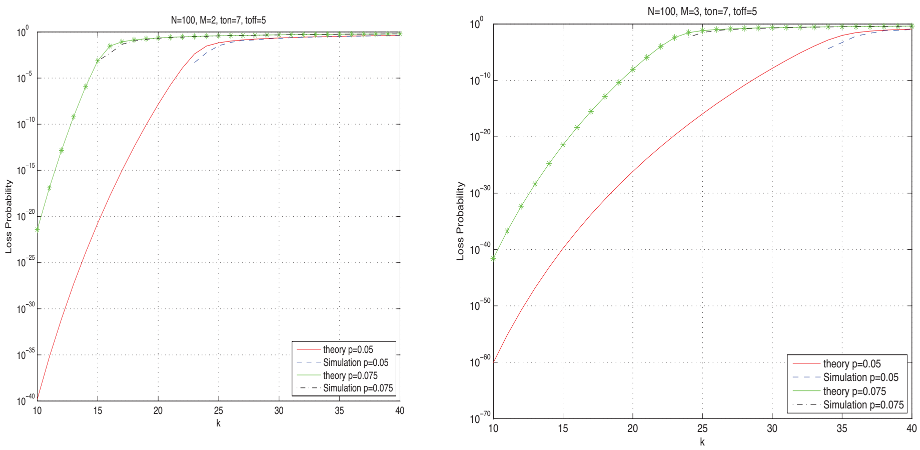


Figure 2.8: Loss Probability (k) for ton=7, toff=5

Simulations points are not plotted for low values of P_L i.e., $P_L < 10^{-5}$, as we did not observe a statistically significant number of missed events, given the simulation time and the values assumed for the other parameters. In order to observe those values we should increase the number of time slots simulated. This would increase the simulation time exponentially, taking several hours in each case. However, we observe a good match in other scenarios.

Figures 2.9, 2.10 and 2.11 compare the probability of *event-loss* P_L, depending on p, between theory and simulations. Again, we show the cases for $M = 2$ and $M = 3$ for the different values we took for the rest of parameters. Note the difference between the cases with $k = 10$ and $k = 20$, specially in the case when T_{on} is smaller. A node with $k = 20$ would be unsustainable.

Figure 2.12 compares the results using the derived loss probability with the previous model presented

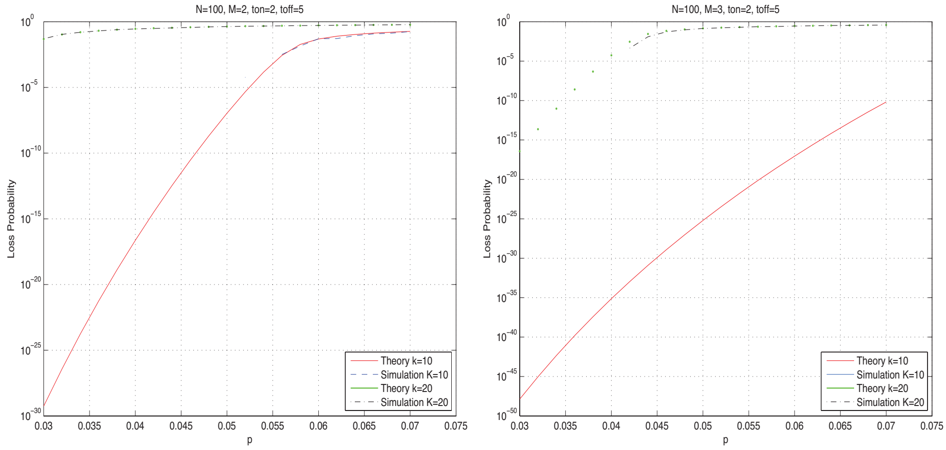


Figure 2.9: Loss Probability (p) for ton=2, toff=5

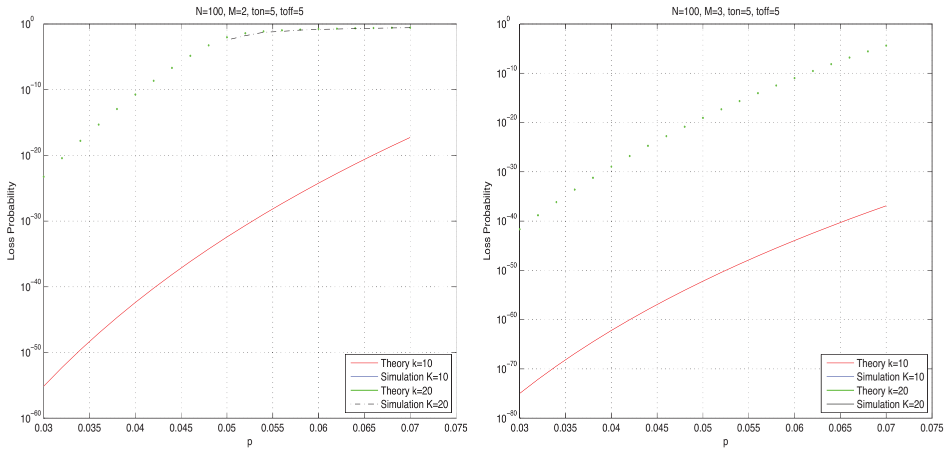


Figure 2.10: Loss Probability (p) for ton=5, toff=5

in [21]. We represent the comparison depending on k and p. We see a good match when k and p increase. However, remind that they are using a slightly different energy model, being able to use the energy they are harvesting in the same time slot. Notice that we only use $M = 1$ because their model was designed for one EH board.

Figure 2.6 presents the P_L, eq. 2.17, versus N, for different k, w and r. As expected, larger battery capacities lead to significant drop in the loss probability. Thus, our model could identify battery specifications, depending on desired loss probability we want to achieve and the other device-specific parameters of the system. Finally, figure 2.6 shows the average time before *event-loss*, eq. 2.27, for various scenarios depending on which battery state we start with.

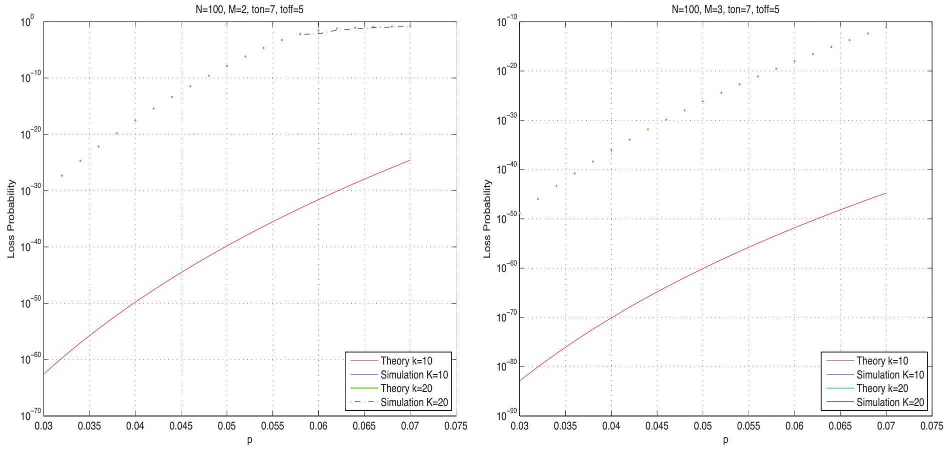


Figure 2.11: Loss Probability (k) for ton=7, toff=5

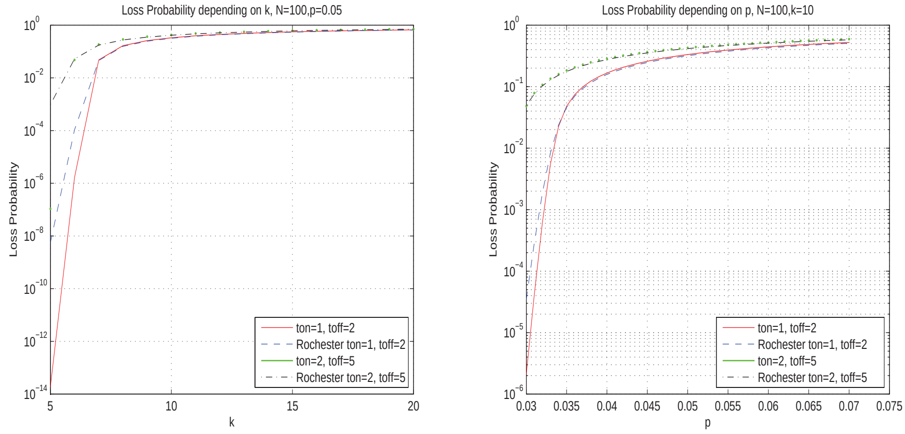


Figure 2.12: Loss Probability comparison with previous model

2.7 Conclusions

In this chapter we have presented MAKERS, a Markov based model for multiple-sources energy harvesting nodes in WSNs. It considers both the number of the harvesting boards attached to the node, as well as the remaining energy of the battery to determine the state of the node. Closed form solutions for the *event-loss* probability and average time before event-loss are derived. The results of our simulations are in good agreement with the derived closed for these parameters. Through our prediction models, the network designer can set the requirements for sensor nodes with many EH-boards, such as the battery capacity. Also, the average time before *event-loss* can be taken into account in future energy management techniques, such as duty cycles, routing paths, among others.

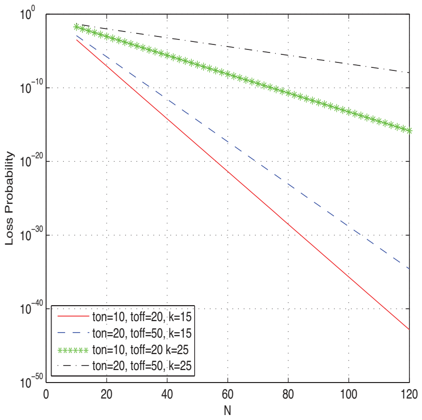


Figure 2.13: Loss Probability (N) for M=2, p=0.02

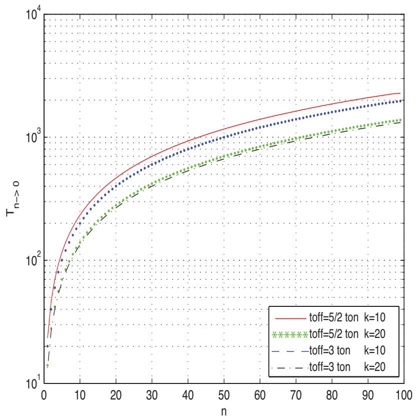


Figure 2.14: Average time before Event-Loss (n) for N=100, M=2, p=0.05

Chapter 3

MAKERS Model Modifications

3.1 Introduction

Modeling sensor networks usually implies taking some assumptions to keep the model simpler to analyze. However, those assumptions are sometimes difficult to justify. Hence, in this chapter, we try to extend the model presented in the previous chapter 2 by adding new parameters that complicate the model while making it more real at the same. These are the changes that we present in this chapter:

- First, in Section 3.2, we propose a modification to account for different kind of energy sources. One source may be normally not present, vibrations that produce energy, for example, while others, like solar energy is present most part of the day in a sunny day. Using different sources in the same node can be, therefore, very beneficial.
- Second, in Section 3.3, we consider the possibility of two energy harvesting boards, from the same source or from different ones, to be dependent, i.e. the state changes of one of them depends on the changes of the other.
- Third, in Section 3.4, MAKERS model assumed no consumption during the sensor idle or sleeping state, when no event occurred. In this chapter, we present how considering that the node consumes some energy changes the model.
- Forth, in Section 3.5, we discuss about adopting a more complex harvesting model.

The rest of the sections in this chapter describe the modifications we just mention. In Section 3.6, we analyze the results of the simulations taken for the different changes made in the model and, eventually, conclude our work in Section 3.7.

3.2 Model for many different sources

In our harvesting model, energy from different sources can be obtained by the same node. Therefore, we introduce a modification of MAKERS in order to work with boards with different features, in terms of average rate of energy harvested and probabilities to switch their state. We first discuss a simplified case of two different sources, and then show how this case can be extended for η different sources.

3.2.1 Model for two different sources

For the two source model, our approach closely follows the analysis for the earlier model defined in Section2.3. Using the same notation, we have:

- *Board A*: ρ_A, r_A, w_A, $\mu_A = \frac{w_A}{r_A+w_A}$
- *Board B*: ρ_B, r_B, w_B, $\mu_B = \frac{w_B}{r_B+w_B}$

We define ρ_B as $\rho_B = b\rho_A$, where b is a real positive number. We keep the definition of the parameters of the model, but with one new constraint of $b+1 \leq k$. The transition probabilities for the general model, $p_{a,b/i,j}$ where $i = 0,A,B,AB$ and $j = 0,A,B,AB$ expressing which the *active* boards in the current slot and future respectively, change as follows:

- $p_{n,n-1/0,j} = p\delta_{0,j}$
- $p_{n,n-1/A,j} = p\frac{k-1}{k}\delta_{A,j}$
- $p_{n,n-1/B,j} = p\frac{k-b}{k}\delta_{B,j}$
- $p_{n,n-1/AB,j} = p\frac{k-(b+1)}{k}\delta_{AB,j}$
- $p_{n,n/0,j} = (1-p)\delta_{0,j}$
- $p_{n,n/A,j} = [(1-p)\frac{k-1}{k} + p\frac{1}{k}]\delta_{A,j}$
- $p_{n,n/B,j} = [(1-p)\frac{k-b}{k} + p\frac{b}{k}]\delta_{B,j}$
- $p_{n,n/AB,j} = [(1-p)\frac{k-(b+1)}{k} + p\frac{(b+1)}{k}]\delta_{AB,j}$
- $p_{n,n+1/0,j} = 0$
- $p_{n,n+1/A,j} = (1-p)\frac{1}{k}\delta_{A,j}$
- $p_{n,n+1/B,j} = (1-p)\frac{b}{k}\delta_{B,j}$
- $p_{n,n+1/AB,j} = (1-p)\frac{(b+1)}{k}\delta_{AB,j}$

	j				
		AB	B	A	0
i	AB	$(1-r_A)(1-r_B)$	$r_A(1-r_B)$	$(1-r_A)r_B$	r_Ar_B
	B	$w_A(1-r_B)$	$(1-w_A)(1-r_B)$	w_Ar_B	$(1-w_A)r_B$
	A	$(1-r_A)w_B$	r_Aw_B	$(1-r_A)(1-w_B)$	$r_A(1-w_B)$
	0	w_Aw_B	$(1-w_A)w_B$	$w_A(1-w_B)$	$(1-w_A)(1-w_B)$

Table 3.1: $\delta_{i,j}$ for 2 different sources

Note that these probabilities are similar to the ones presented before, but they express the difference that board B can harvest at a different rate than A. For the special cases of the two boundary states with energy level 0 and $N-1$ we find the following differences:

- $p_{0,0/0,j} = \delta_{0,j}$
- $p_{0,0/A,j} = \frac{k-1}{k}\delta_{A,j}$
- $p_{0,0/B,j} = \frac{k-b}{k}\delta_{B,j}$
- $p_{0,0/AB,j} = \frac{k-(b+1)}{k}\delta_{AB,j}$
- $p_{0,1/0,j} = 0$
- $p_{0,1/A,j} = \frac{1}{k}\delta_{A,j}$
- $p_{0,1/B,j} = \frac{b}{k}\delta_{B,j}$
- $p_{0,1/AB,j} = \frac{(b+1)}{k}\delta_{AB,j}$
- $p_{N-1,N-1/0,j} = (1-p)\delta_{0,j}$
- $p_{N-1,N-1/A,j} = [(1-p)+p\frac{1}{k}]\delta_{A,j}$
- $p_{N-1,N-1/B,j} = [(1-p)+p\frac{b}{k}]\delta_{B,j}$
- $p_{N-1,N-1/AB,j} = [(1-p)+p\frac{(b+1)}{k}]\delta_{AB,j}$

We decided not to include a figure for this case because of the increase of number of transition lines, which affects the general understanding. We encourage the reader to review figures 2.3 and 2.4 and observe the probabilities we just presented to figure out how this figure would be. In table 3.1, all the possible cases for $\delta_{i,j}$ for the case of 2 different sources, A and B, can be observed.

		j			
		AB	B	A	0
i	AB	$(1-r_A)(1-nrr_B)$	$r_A(1-rr_B)$	$(1-r_A)nrr_B$	$r_A * rr_B$
	B	$w_A(1-wr_B)$	$(1-w_A)(1-nwr_B)$	$w_A * wr_B$	$(1-w_A)nwr_B$
	A	$(1-r_A)nrw_B$	$r_A * rw_B$	$(1-r_A)(1-nrw_B)$	$r_A(1-rw_B)$
	0	$w_A * ww_B$	$(1-w_A)nww_B$	$w_A(1-ww_B)$	$(1-w_A)(1-nww_B)$

Table 3.2: $\delta_{i,j}$ for 2 not independent different sources

This new model can be simplified in a similar manner as the model presented in Section 2.3.3. The parameter α is re-visited, giving a new form that will let us compute the derived formulae:

$$\alpha = \frac{1}{k}\mu_A(1-\mu_B) + \frac{b}{k}(1-\mu_A)\mu_B + \frac{b+1}{k}\mu_A\mu_B \tag{3.1}$$

Here, the first term of the summation represents the case when only board A is *active*, the second when only board B is *active*, and finally, the third term gives the case for both boards being *active*. This expression can be trivially extended using different energy harvesting rates ρ_i for a board i, assuming that ρ_i is a multiple of the base rate ρ_A, i.e., $\rho_i = \psi\rho_A$, $\psi > 0$. Along the same lines, α is easily adapted to consider all the cases of each of the combination of boards being in the *active* state, as shown in the two board example in eq 3.1.

3.3 Model considering two not-independent boards

The possibility to assume that two energy harvesting boards are working in an independent manner is not always feasible because it may be some connection between the different sources we are harvesting from. Even more when two different boards get the energy from the same source, e. g. two solar panels. Therefore, in this subsection we analyze how the parameters of the system change when we consider two boards which their probabilities to switch states are not independent. As we did before, we consider board A, as the main one, and the parameters of the second one, B, depend on the ones of A. We define some new probabilities for the state switches of B, conditioned to the changes occurred in the board A, which can be observed in table 3.2, defining the new $\delta_{i,j}$ for this case.

We would like to describe some of the new parameters that appear in table 3.2. For example, if we observe the transition from *AB active* to just *A active* ($(1-r_A)nrr_B$), nrr_B denotes the probability of board B to switch from *active* to *inactive* conditioned that A keep its state to *active*. Similarly, observing the transition from A *active* to B active ($r_A * rw_B$), rw_B is the probability of B to change from *inactive* to *active*, knowing the fact that A switched from *active* to *inactive*.

This dependency between the transition probabilities of A and B change equation 3.1, because different probabilities combining the states of A and B cannot be define as before. Next we analyze the changes made in this case, as observed in 3.2. Defining μ_{AB} as the probability of A and B being *active*, μ_B for the case of B *active* and A *inactive*, μ_A for A *active* and B *inactive*, and, finally, μ_0 for both boards being *inactive*. These probabilities will redefine equation 3.1 as follow:

$$\alpha = \frac{1}{k}\mu_A + \frac{b}{k}\mu_B + \frac{b+1}{k}\mu_{AB} \tag{3.2}$$

Using the next system of equations, we can find the probabilities needed for this new α:

$$\begin{bmatrix} \mu_{AB} \\ \mu_B \\ \mu_A \\ \mu_0 \\ 1 \end{bmatrix} = \begin{bmatrix} (1-r_A)(1-nrr_B) & w_A(1-wr_B) & (1-r_A)nrw_B & w_A * ww_B \\ r_A(1-rr_B) & (1-w_A)(1-nwr_B) & r_A * rw_B & (1-w_A)nww_B \\ (1-r_A)nrr_B & w_A * wr_B & (1-r_A)(1-nrw_B) & w_A(1-ww_B) \\ r_A * rr_B & (1-w_A)nwr_B & r_A(1-rw_B) & (1-w_A)(1-nww_B) \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \mu_{AB} \\ \mu_B \\ \mu_A \\ \mu_0 \end{bmatrix} \tag{3.3}$$

We will be able to compute the equations derived in Sections 2.4 and 2.5 using the parameter α in equation 3.2 and solving the system in 3.3.

3.4 Model assuming idle energy consumption

Node consumption during the idle/sleeping state is difficult to analyze. Usually, we cannot consider it negligible as batteries have some leakage and some energy is necessary to maintain the node on, even though it is not sending or reporting anything. Hence, in this section we discuss the MARKERS model considering that the node spends also energy when an event does not occur. We assume this idle consumption is inferior to the one when an event happens, as transmission is the major cause of energy consumption in a sensor node. Therefore, defining the consumption rate as c_{idle}, parameter l compares this rate with the consumption of one event, E:

$$l = \frac{E}{c_{idle}T} \tag{3.4}$$

Where $l > 1$ and $l \in R$, to illustrate the event consumption to be bigger. In order to build the model in a similar manner as the one made in section 2.3, we are going to start with a model with larger number of states to finally reduce it to analyze it. In figure 3.1 we can see an example for $M = 2$, $k = 2$ and $l = 3$, we observe the model with $(M+1)klN$ states. For the sake of simplicity, we only include the transition probabilities when an event does not occur, because the rest, when an event runs, are similar to the ones in the $(M+1)kN$ states model, figure 2.1.

Observing figure 3.1, if an event does not happen when we are in state n with 2 *active* boards, the node harvests $\frac{2}{2}E = E$ and consumes a total of $\frac{1}{3}E$. Therefore, the future energy state is $n+4/6$. We

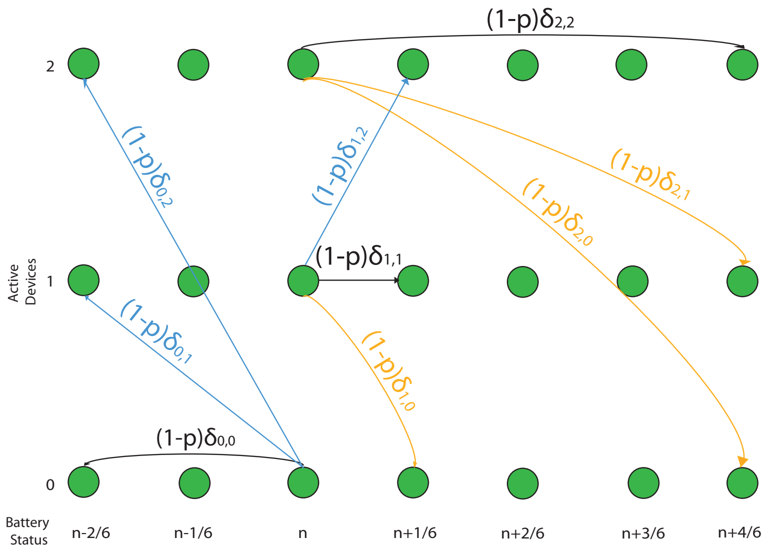


Figure 3.1: (M+1)klN states model with idle consumption

observe, three transitions because the states of the different boards may change. Again, from the case of 1 *active* board in n, node harvests $\frac{1}{2}E$ and consumes $\frac{1}{3}E$ when no event happens. Hence, the next energy state is $n+1/6$. Finally, when all boards are *inactive*, the node consumes $\frac{1}{3}E$ and transitions to energy state $n-2/6$. Generally, we can state that the next energy level, having i *active* boards and starting from n is: $n+\frac{i}{k}-\frac{1}{l}=n+\frac{il-k}{kl}$.

Note that depending on the values of k and l, energy consumed can be larger than energy harvested not only in the case when any board is *active*. Consequently, in the general case of $(M+1)N$ states, more states would not be able to reach a higher state in terms of energy. Following parameters, β_i and β_i', will help us set the transition probabilities in the model:

$$\beta_i = \begin{cases} \frac{il-k}{kl} & il-k>0 \\ 0 & otherwise \end{cases} \tag{3.5}$$

$$\beta_i' = \begin{cases} -\frac{il-k}{kl} & il-k<0 \\ 0 & otherwise \end{cases} \tag{3.6}$$

After observing the example shown in figure 3.1 and the parameters defined in equations 3.5 and 3.6 , we can develop the general case of $(M+1)N$ states which can be observed in figure 3.2. The transition probabilities are defined as follows, when $i>0$:

- $p_{n,n-1/i,j} = [p\frac{k-i}{k} + (1-p)\beta_i']\delta_{i,j}$
- $p_{n,n/i,j} = [(1-p)(1-\beta_i'-\beta_i) + p\frac{i}{k}]\delta_{i,j}$

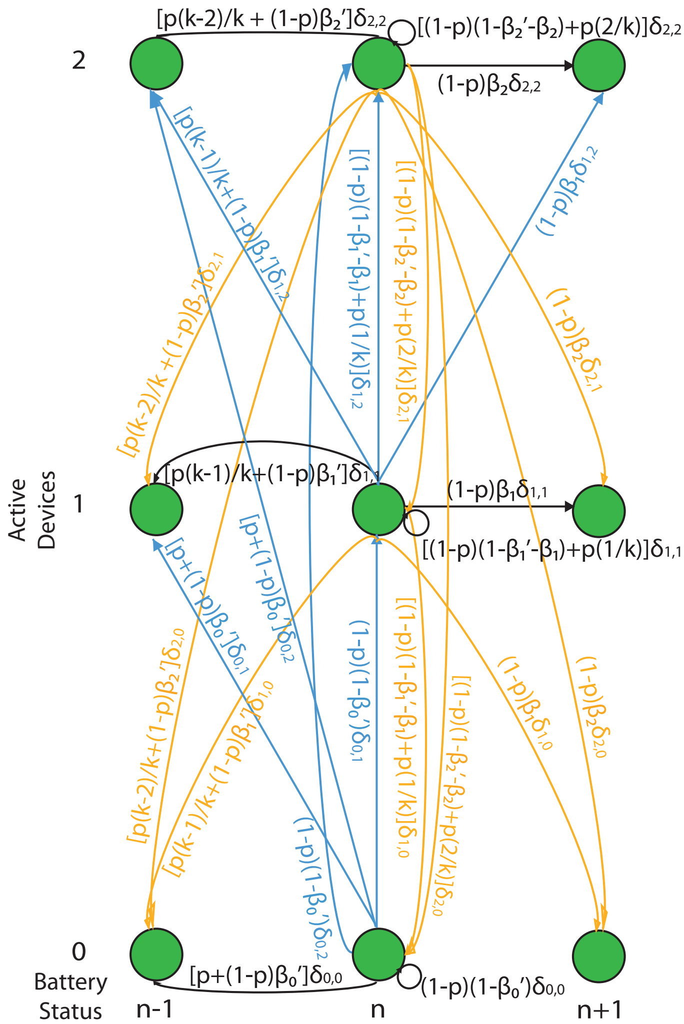


Figure 3.2: (M+1)N states model with idle consumption

- $p_{n,n+1/i,j} = (1-p)\beta_i\delta_{i,j}$

Note that $\beta_0 = 0$ in any case, it is not possible to reach a higher level in the case of $i = 0$:

- $p_{n,n-1/0,j} = [p + (1-p)\beta_0'\delta_{i,j}$
- $p_{n,n/0,j} = (1-p)(1-\beta_0')\delta_{i,j}$
- $p_{n,n+1/0,j} = 0$

In 3.3, we see the extreme states, i.e. with energy $n = 0$ and $n = N-1$, for the $(M+1)N$ states model. We have assumed that even though the node has less than $\frac{1}{l}E$ energy and does not harvest sufficient energy to keep a positive balance, it remains in the same state. An alternative is to consider the

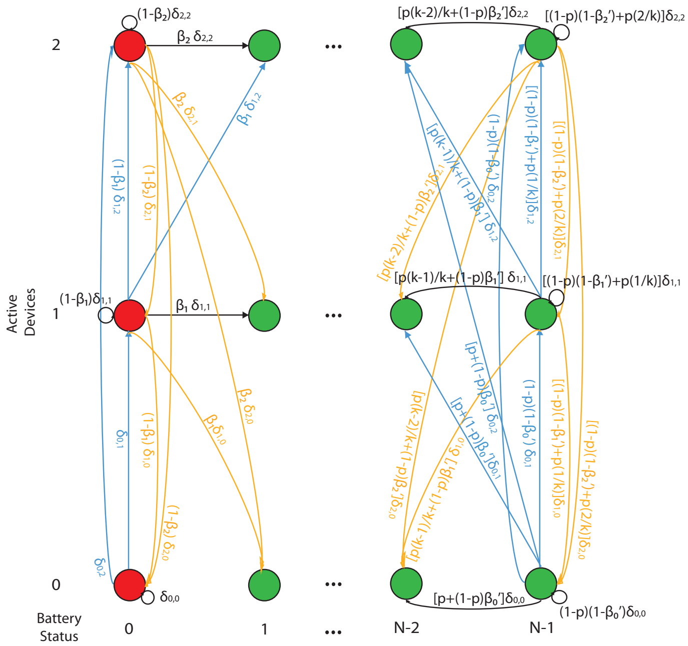


Figure 3.3: (M+1)N states model with idle consumption, extreme states

state in this case as absorving, specifying that once the node reaches it, it fails and would have to have its battery replaced.

Similarly to the discussion in section 2.3, it is possible to simplify to a N states. Using the same parameter α defined then and the next ones, β and β', the transition probabilities are simplified:

$$\beta = \sum_{i=0}^{M} \beta_i \phi_i \tag{3.7}$$

$$\beta' = \sum_{i=0}^{M} \beta_i' \phi_i \tag{3.8}$$

As we see in figure 3.4, the different transition probabilities are revisited:

- $p_{n,n-1} = p(1-\alpha) + (1-p)\beta'$
- $p_{n,n} = p\alpha + (1-p)(1-\beta'-\beta)$
- $p_{n,n+1} = (1-p)\beta$

Finally, for the extreme states, $n = 0$ and $n = N-1$:

- $p_{0,0} = 1-\beta$
- $p_{0,1} = \beta$

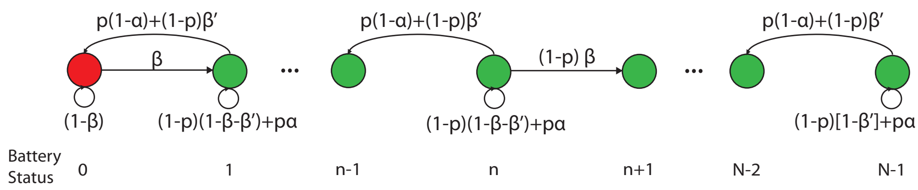


Figure 3.4: Simplified N states model with idle consumption

- $p_{N-1,N-2} = p(1-\alpha)+(1-p)\beta'$
- $p_{N-1,N-1} = p\alpha+(1-p)(1-\beta')$

Following a development similar to the one seen in Section 2.4, the parameter γ is revisited and can be defined as follows:

$$\gamma = \frac{(1-p)\beta}{p(1-\alpha)+(1-p)\beta'} \tag{3.9}$$

While the *event-loss* probability keeps the same structure as the previous times, the changes in the Markov chain model make the Average Time before Event-Loss adopt this new form:

$$T_{n\to 0} = \frac{1}{p(1-\alpha)+1-p)\beta'} \sum_{y=1}^{n} \sum_{x=0}^{N-1-y} \gamma^x \tag{3.10}$$

3.5 Model considering a more complex harvesting model

Our harvesting model assumes that when a energy harvesting board is *active* harvests energy at a constant rate, what we call average rate. However, usually harvesting rates vary highly during time because sources from where we are getting energy also do. For example, when considering a solar panel, the rate changes considerably depending on the weather. For the same location and node, a sunny summer day will be total different from a snowy day in winter.

MAKERS model can be observed from another point of view. Instead of having M harvesting boards, only one is considered. This unique board harvest energy from a given source and its harvesting rate varies enough that cannot to be assumed as an average constant, only during a sufficient small time, the discrete time unit, T. If we take ρ as the minimum energy rate the board harvests when is *active* and $M\rho$ as the maximum possible rate, it is possible to adapt MAKERS to account for a more detailed harvesting distribution.

The board can harvest at M different levels. As the separation between levels, ρ, is small we can assume that when it is harvesting, it will do it in one particular level. The concept of the model remains the same, k is $\frac{E}{\rho T}$ and likely to be large, as ρ is small. Remind that $k \geq M$, to maintain the definitions of the equations and transition probabilities.

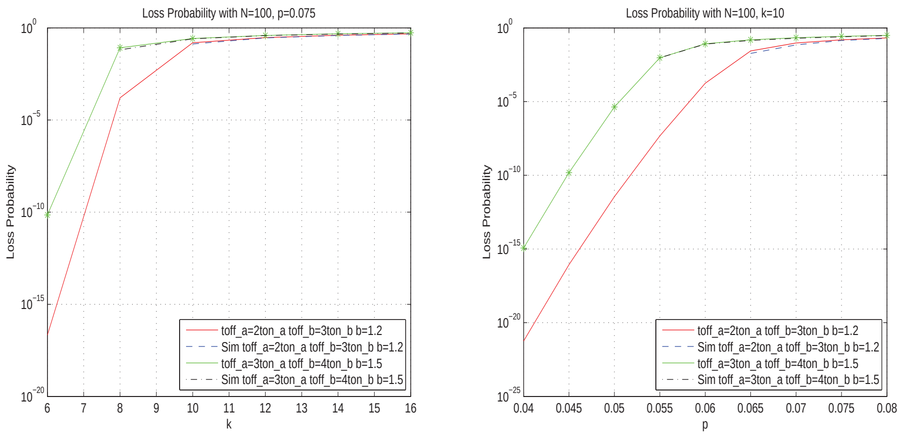


Figure 3.5: Loss Probability N=100

The more complex analysis here is to set the probabilities of the board to be harvesting at each level: ϕ_i for $0 \leq i \leq M$. Observing the energy harvested during a sufficient amount of time, from days to months, depending on the source and the changes in the environment could give us enough information to approximate these probabilities.

3.6 Results

Figures 3.6 and 3.6 compare the *event-loss* probability derived with those obtained from the simulations in the case of different boards, harvesting from different energy sources or from the same one but having different features in terms of efficiency, for example. In the first figure, we compare the results by varying k, and later we vary p in the second one. Every harvesting board has different parameters exponentially distributed with different mean time for the state durations. We took situations where T_{on} is smaller than T_{off}, in both boards. Board b harvest slightly more than board a when it is in the *active* state.

Figures 3.6 and 3.6 show the *event-loss* probability, both the result from the formulae and the one simulated for thederived with those obtained from the simulations. As we did in the previous modifications, we first compare the results by varying k and later we vary p. We observe two cases in each figure that only change in the term of the consumption during the idle state, when no event occurs. We took higher values for this parameter, $\frac{1}{9}E$ and $\frac{1}{10}E$, in order to realize the difference it makes this small change between both cases. However, it is also possible to analyze a node that is usually listening to the neighbors and when receives a packet, it forwards it to another node or to the final destination.

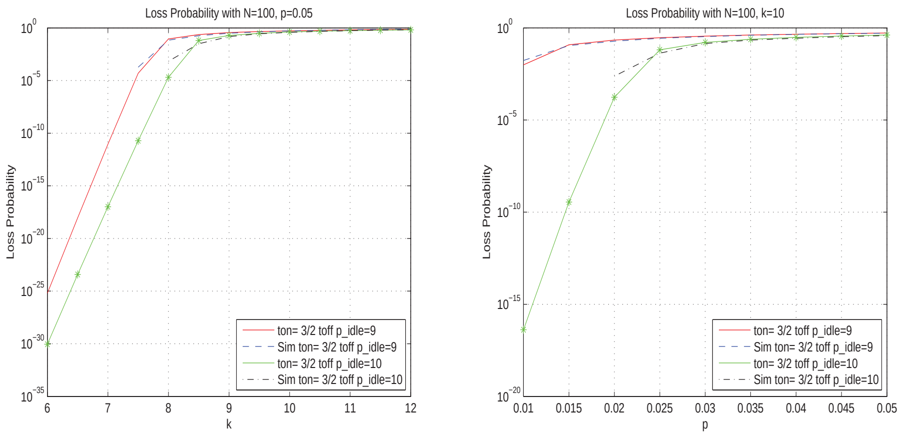


Figure 3.6: Loss Probability N=100

3.7 Conclusions

In this chapter, we have extended the MAKERS model: We have extensively developed the model with different energy sources for multiple energy harvesting boards; we have discussed the case of having a board which parameters depend on the ones of another board, typically for cases that obtain the energy from the same source; we have dropped the assumption of no consumption during the time that the node is not sending a packet which leads to a better energy model; and, finally, we talk about how the harvesting model could be extended further.

We have added some new parameters that if combined help the model to be more realistic and close to real scenarios. However, deeper study should be necessary improving the current harvesting model, differencing it depending on the characteristics of every specific energy source.

Chapter 4

MAKERS' Experiments

4.1 Introduction

In this chapter we analyze the results of the experiments we realized to prove the validity of the proposed MAKERS model. We used the equipment available in our laboratory to compare a real case of sensor nodes powered by a RF energy harvesting board with the theoretical predictions. We study the time a node needs to run out of energy, and consequently lose an event, on such conditions.

The rest of this chapter is outlined as follows: In Section 4.2, we describe the environment and equipment used in the experiments. We talk about how we conducted the measurements in Section 4.3. In Section 4.4 we discuss the assumptions we had to do and their implications.In Section 4.5, we evaluate the performance results of the model comparing it to real cases. Finally, in Section 4.6, we conclude our work.

4.2 Setup

In order to conduct this set of experiments we used MICA2 sensor nodes developed by Xbow and the University of California, Berkeley; [16]. The nodes where programed using Tiny OS, an open source component-based operating system and platform targeting wireless sensor networks, written in the nesC programming language; [17]. Both the transmitter, the node we were analyzing, and the receiver, the one that acted as base station, were MICA2 nodes.

We used the RF harvesting kit shipped by Powercast, [4], to feed the transmitter. This kit consists in a power transmitter, at a constant of 3W, and a harvesting board that collects energy received by an antenna and charges a capacitor or some batteries. We used a 220 mF capacitor as energy supplier for the transmitter. The receiver is not energy constrained since it was connected to a laptop through usb to charge and collect the data.

In figure 4.1 we can see the working environment of our experiments. The power transmitter was located 43 inches away from the harvesting board.

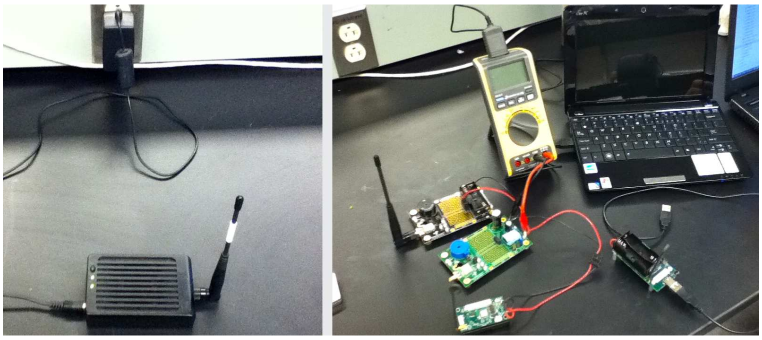


Figure 4.1: Setup done for the experiment

4.3 Experiment

We first observed the harvesting process measuring the voltage charging of the capacitor. Using a multimeter, we took measures every $0,5$ seconds. Although the power transmitter was on during the whole harvesting process, the capacitor couldn't increase its residual voltage during all the time slots. Therefore, we considered those periods when no energy was harvested as *inactive* mode periods for the harvesting board and we computed the average voltage gained during the rest of the measures. Recall that the energy stored in a capacitor can be computed as:

$$E_c = \frac{1}{2}CV^2 \tag{4.1}$$

The analyzed node, the transmitter, was programmed to send packets to a base station with the current time and the remaining voltage. The size of the packet is 40 B and the theoretical data rate is 19200 bps. However, the node needs more time to transmit a packet because of the internal process. When the node is not sending any package, it switches the radio off, saving a considerable amount of energy during those periods.

The event times were pre-computed in Matlab following a random exponential distribution since TinyOS does not support exponential variables, only uniform distributed.

We measured the power consumption during both modes by programming the node first to work only in the currently analyzed mode. We charged the capacitor using the RF harvesting board and when the capacitor arrived to its maximum capacity, we turned off the power transmitter and switched the node to on, so it did not interfere. While observing the discharging on the transmission mode, we could measure the time between packets, since we programmed the node to continuously send packets. We use this packet time as the time slot unit for our model.

We computed the average voltage consumption and later the average energy consumption using eq. 4.1. Using these values we could calculate the parameters k and l of the model.

The energy storage size, N, was derived as follows: the operating voltage range of the MICA2 node is

$E_{harvest}$	$4,0366$ nJ/s	**k**	$13,2639$
E_{Tx}	$710,1$ nJ/s	**l**	$5,7566$
E_{idle}	$21,431$ nJ/s	**N**	579
μ	$0,7274$ μ J/s	**Ts**	$0,031$ s
p 8	$0,0039$	**p 16**	$0,0021$

Table 4.1: Experimental parameters

$1,8$ to $3,3$ volts; hence we divided the range by the average voltage consumption of a packet transmission at full power to obtain N. However, we observed that the node runs out of energy approximately in the range $1,825$ to $1,83$. Therefore, $n = 0$ equals to have a voltage just below to $1,83$ V, forcing the node to die.

The values of the parameters used in the experiments can be found in Table 4.1. We used two different average time between events, 8 and 15 seconds, so we have two different event probability values.

4.4 Assumptions

First, during the harvesting/charging measurements, we can not measure exactly the power receiver at the antenna of the harvesting board nor the efficiency of the converting it to a profitable power for the capacitor. The transmitter is supposed to transmit at 3W continuously, we located the node not far from it, but real working scenarios would not be able to work in these conditions.

We decided to just observe the time the node needs to consume all its energy and, consequently, be unable to transmit any packet. When the node dies, the capacitor continues to draw energy which makes it difficult to ever return to the range where the node operates. Additionally, the node should have to be programmed to switch to sleep just before it runs out of energy and set a timer to wake up, to return to normal working process. However, the node would have to be slept to much time which would affect the experiments evaluation.

We conducted the experiments 10 times. In the results section, 4.5 we observe the average of all the

4.5 Results

Figure 4.5 shows the voltage charging process of the capacitor used for the experiments. We observe the charging over the voltage operating range of the MICA2 node ($1,8$ - $3,3$). In figure 4.5 we can see the voltage consumption of the MICA2 node using the previous capacitor as power source and maintaining the same energy mode during its operation until it dies. One energy mode is transmission at full power and the other one was also observed when the radio was switched off, the division between the average

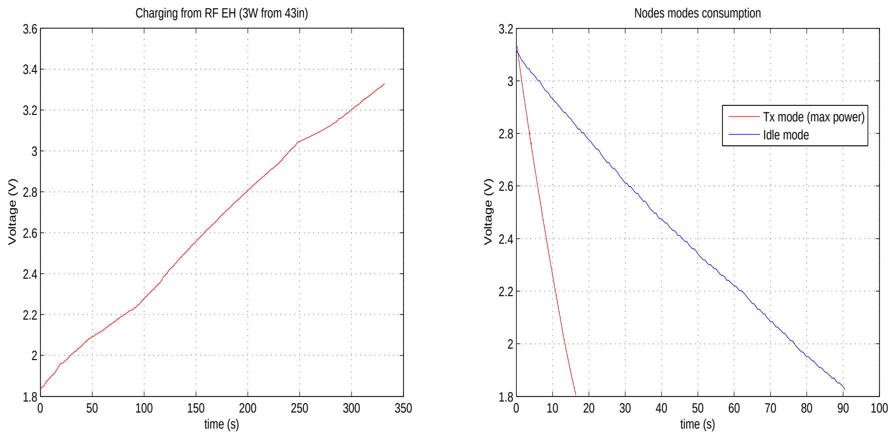


Figure 4.2: Capacitor charging and Node Consumption

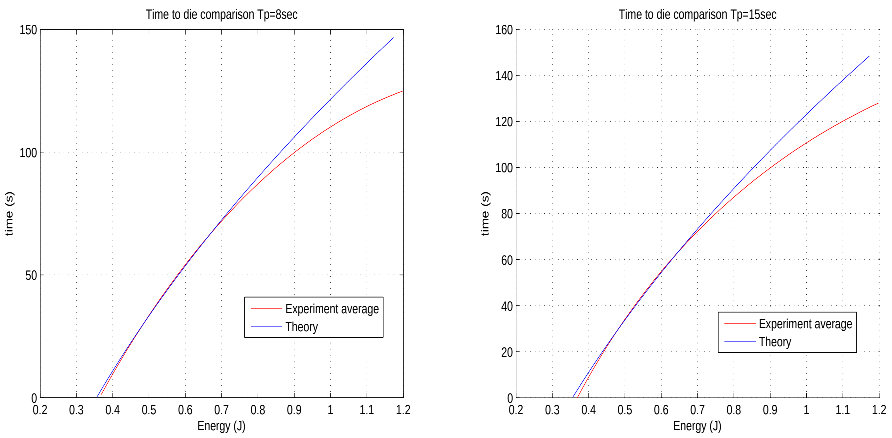


Figure 4.3: Average time before Event-Loss (Energy)

voltage consumption of the transmitting mode and the average in the idle, radio off, mode gave us the l parameter that we see in 4.1.

Figures 4.5 and 4.5 compare the average time before *event-loss*, eq. 2.27, with the average time the device was alive with that amount of energy. They represent the case with average time between events equal to $T_p = 8s$ and $T_p = 15s$, respectively. We observe a good match between the theoretical value obtained from the MARKERS model with the system parameters and the experiments we performed. However, there is not big changes between the two cases, with different average times between events. In 4.4, it is represented the standard deviation of the set of experiments depending on the residual energy in the capacitor. Observe that the deviation increases when we get close to the capacitor full capacity, it is as well, the range where the average time differs most with the value predicted.

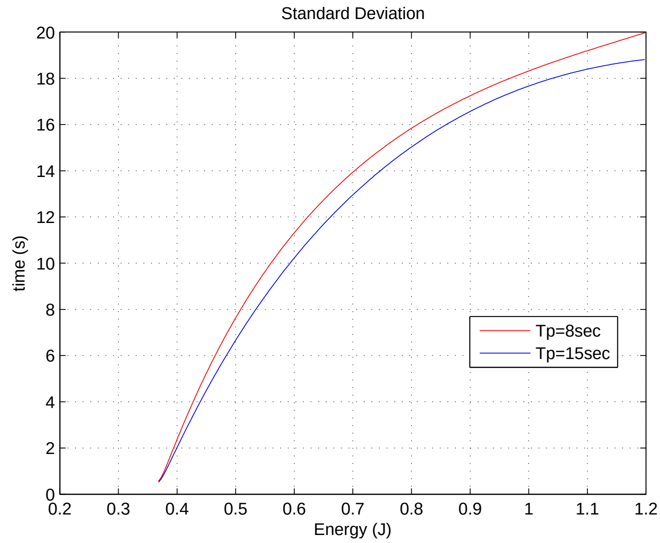


Figure 4.4: Standard Deviation of the experiments

4.6 Conclusions

In this chapter, we have presented the results of out experiments to compare the theoretical predictions performed with the presented MAKERS model. The results show an acceptably good agreement between the real scenario and the derived average time before event-loss.

However, we could not conduct experiments with other energy sources, such as solar. In addition, future work needs to be done to experiment using multiple energy harvesting boards, from the same source or different ones, to finally prove the validity of the model.

Chapter 5

MAKERS Decision Process

5.1 Introduction

The purpose of enabling energy harvesting capabilities to sensor networks is to provide an autonomous and independent working-life to them. It would be very interesting that nodes can adjust themselves in order to work taking into account the changes in their environment being able to extend their operation for a larger time. Therefore, researchers have been working on several models and algorithms that allow nodes to decide how to act to further benefit during their function.

We present an opportunistic transmission strategy for wireless sensor networks that operate in an energy-constrained environment. Our proposed framework is based on the MAKERS model presented in chapter 2. The node features a binary-decision agent that attempts to run an event depending on the battery level, the current harvesting power and the delay since the last event. Therefore, the agent is capable of increasing or decreasing the sensing frequency observing these parameters. The system is modeled as a Markov decision process (MDP), which is used to find optimum threshold for transmission decision.

Extensive simulations are performed to verify the performance of our proposal over energy harvesting wireless sensor nodes with different parameters. The simulation results show that the MAKERS Decision Process scheme has higher efficiency in terms of events successfully transmitted than the random case showed in the previous chapter.

The rest of this chapter is organized as follows: In Section 5.2, we provide some background about Markov Decision Processes. Section 5.3 describes our proposed model. We show and discuss the results of our simulations in Section 5.4. Finally, Section 5.5 concludes this chapter.

5.2 Markov Decision Processes

In this section, we discuss the theory behind the general framework of Markov Decision Processes, some techniques to solve them and, later, we comment some of the previous work using MDP in Wireless

Sensor Networks and other networking environments.

Markov decision process (MDP) is a widely used mathematical framework for modeling decision-making in situations where the outcomes are partly random and partly under control. The aim is to devise a plan, or *policy*, that will maximize the expected benefit of interacting with the environment. Several algorithms have been used to find the optimal policy given a MDP, as a mapping from states to actions, [13]. On-line variants, known as *Reinforcement Learning* techniques that learn the optimal policy through experience with a previously unknown domain have also been considered for researchers during the last decades; [15] [5].

The standard algorithms for MDPs are efficient depending on the number of states and actions within the system. However, the formal requirements of the framework, specially the Markov property, force representations of problems that use a very large number of states. Remind that the Markov property states that of the next state must be a function of the current system state and action only, not any of the previous states.

5.2.1 Basic Terminology

A MDP is a discrete time stochastic control process, formally presented by a tuple of four objects $\{S, A, P_a, R_a\}$.

In the MDP's framework, the environment is divided into states. S is the state space, which, in general, is assumed to be finite. We will refer to the current state as s, where $s \in S$.

A is the action space. It defines the actions that are possible or under consideration in the environment. It may be the case that only a subset of A is possible in any given state, so A_s is used to refer to the set of actions possible in s.

$P_a(s,s')$ is the transition probabilities matrix, that maps from *SxAxS* into different probabilities, in $[0,1]$. Specifically the probabilities that the next state will be s' when action a is taken in the current state s. The value of the current states follows a stochastic process that has the Markov property: the probability of state s_{t+1} having any given value depends only upon state s_t and the decision taken, a_t.

$R(s,a)$ is the immediate reward function that maps from *SxA* to real-valued rewards. It must be finite for all s and a.

Hence, a MDP is seen as an extension of Markov chains, adding actions (choices) and rewards (motivation) to the states. If the actions are fixed, an MDP reduces to a Markov chain.

A policy, π, for an MDP is a scheme for assigning actions in A to states in S. Policies may be *stationary*, which means that they make a simple mapping from states to actions, or they may be *non-stationary* which means that the action assigned to a particular state may change depending upon some other factor. A family of policies that optimize the agent's behavior with respect some optimal criteria is referred to as Π^*. Algorithms for MDPs usually seek for some optimal $\pi^* \in \Pi^*$ or some approximation.

The goal is to maximize some cumulative function of the rewards. A value function, V, is a mapping from elements of S to real values. A value function represents the utility of a state, which is some measure of the reward for continuing in the environment from that state. The policy assigns to each state $s \in S$ the action that maximizes the expected value of s given the values assigned to s and states reachable from s in one step by some value function V. The value function for an optimal policy π^* is called the optimal value function and is denoted by V^*. The following system of nonlinear fixed-point equations, known as the Bellman equation for infinite-horizon MDPs [2]:

$$V(s) = R(s,a) + \gamma \sum_{s' \in S} P_a(s,s')V(s') \tag{5.1}$$

where γ is the discount factor, $0 \leq \gamma \leq 1$. The discount factor determines the importance of future rewards. A factor of 0 will make the agent "opportunistic" by only considering current rewards, while a factor approaching 1 will make it strive for a long-term high reward.

If the transition probabilities are well known, π^* can be solve by value or policy iteration and it becomes a straightforward computational problem. However, if those probabilities are unknown, then this is a problem for reinforcement learning. In RL, the learning agent interacts with the environment collecting rewards as it tries out different actions in different situations. This experience should help it decide which actions are better long term consequences in terms of maximizing the rewards obtained. Next we briefly review some techniques and algorithms to solve MDPs that we will use in our simulations.

5.2.2 Solving Markov Decision Processes

Value Iteration

One way, then, to find an optimal policy is to find the optimal value function. The following simple algorithm called *value iteration*, from [2] determines V values that converge to the correct optimal V^* values:

initialize $V(s)$ arbitrarily
loop until policy good enough
    loop for $s \in S$
        loop for $a \in A$
            $Q(s,a) := R(s,a) + \gamma \sum_{s' \in S} P_a(s,s')V(s')$
        $V(s) := \max_a Q(s,a)$
    end loop
end loop

The computational complexity of the value-iteration algorithm per iteration, is quadratic in the number of states and linear in the number of actions. It is difficult to determine when to stop the value iteration algorithm. For example, authors in [19] propose to stop the algorithm once achieved a greedy policy. It says that if the maximum difference between two successive value functions is less than ϵ, then the value of the greedy policy (the one obtained by choosing, in every state, the action that maximizes the estimated discounted reward, using the current estimate of the value function) differs from the value function of the optimal policy by no more than $2\varepsilon\gamma/(1-\gamma)$ at any state. This provides an effective stopping criterion for the algorithm.

Policy Iteration

The *policy iteration* algorithm works modifying directly the policy instead of finding the optimal value function. It looks as follows:

choose an arbitrary policy π'
loop
  $\pi := \pi'$
  compute the value function of policy π:
    solve the linear equations
      $V_\pi(s) = R(s,\pi(s)) + \gamma \sum_{s' \in S} P(s,\pi(s),s') V_\pi(s')$
  improve the policy at each state:
    $\pi'(s) := \arg\max_a (R(s,a) + \gamma \sum_{s' \in S} P(s,\pi(s),s') V_\pi(s'))$
until $\pi = \pi'$

The value function of a policy is just the expected infinite discounted reward that will be gained, at each state, by executing that policy. Once we know the value of each state under the current policy, we consider whether the value could be improved by changing the first action taken. If it can, we change the policy to take the new action whenever it is in that situation. When no improvements are possible, then the policy is guaranteed to be optimal.

5.2.3 Related Work

Researchers have done some previous work in the area of wireless sensor networks using markov decisions processes. Munir et al. [9] proposed an application-oriented dynamic tuning methodology for WSNs based on MDPs) performing dynamic voltage, frequency, and sensing (sampling) frequency scaling (DVFS2). Traditional microprocessor-based systems use dynamic voltage and frequency scaling (DVFS) for energy optimizations. Authors added sensing frequency tuning to this approach to meet application requirements because the sensed data delay depends upon the sensor node sensing frequency as it dictates the amount of processed and communicated data.

Van Phan et al. [18] consider an opportunistic transmission strategy, named binary-decision based transmission (BDT), for WSNs hosting delay-sensitive applications. Their transmission scheme attempts to transmit only under good channel conditions whenever possible. By exchanging the control messages for channel measurement and associated feedback, the receiver can measure the channel quality and the sender can retrieve such information piggybacked in the return message. Therefore, they use a MDP-based framework to obtain the optimum threshold for successful transmission in the BDT.

Authors in [12] investigate the performance of power selecting techniques in sensor networks with energy harvesting. The sensors can choose from a set of available transmission modes with each scheme consuming a different amount of energy. The packet error probability decreases in function of the energy used on transmission. The problem of selecting the power level at which a sensor should transmit is formulated as a MDP and the performance of the transmission policy thus derived is compared with that of an energy balancing policy as well as an aggressive policy.

Huijiang et al. [8] present a model where an energy harvesting sensor node is able to choose if it sends its data straight to the destination or relies on a neighbor sensor, the relay, to help it transmit the data to the sink. This last approach helps the node to achieve a similar bit error rate using lower transmission power. In order to optimally determine through a MDP if the relay should be used or not at a given time, in addition to its own state information, the source also needs to know the information at the relay. Although they exchange data during transmission, in periods without data, communicate the state information of the relay in real time represents a significant overhead. Thus the node may have to take may have to base its decision on stale state information. Therefore, authors focus on obtaining the optimal information using only a part of the system information.

5.3 MAKERS Decision Process

System State definition

This model is based on the fundamentals of the MARKERS' model described in chapter 2. Therefore, we have a sensor node with several energy harvesting boards connected to it. As we did previously, we start from a model with a total of M EH boards, that harvest energy from the same source.

The current node state is defined by a tuple of three parameters: $S = \{n, m, \tau\}$. n is the battery level, representing the amount of residual energy in the battery storage and cap take values from 0 to $N-1$. m is the number of active energy harvesting boards, from 0 to M, which is the total number of boards. Finally, τ, represents the number of time units since the last successful event, $\tau \in \{0, ..., T-1\}$. T is the desired delay between events. The purpose of including this new parameter is to remove part of the randomness of the event occurrence. The larger τ is, the bigger the node's likelihood to run an event is. The model can be seen as a way to determine the sensing and transmitting frequency depending on the current state. τ depends on the time of the last successful event because, here, we assume that the channel is not ideal and p_l is the probability of a sensing report to have errors and therefore, need to be send again.

Remind from the original MARKERS model, that an event consumes a total amount of energy equal to E. Therefore, the battery or super-capacitor storage capacity is $(N-1)E$.

The total number of states is $N*(M+1)*T$. Next, we will define the action set.

Action set

The system we are modeling is based on a node reporting sensing events to a sink. Hence, the agent featured by the node will be able to decide between two possible actions: stay idle or run an event. We define the action to take as a, which can be either 0, for not running an event, or 1 when doing so.

Both actions are available in all the possible system state, except that sensing events are not possible when there is not enough energy to run them. Thus, action $a = 1$, is not possible in states where $n = 0$.

Transition probabilities

The transition probabilities, $p_a(s,s')$, to the future state depend on the current one as well as the decision taken. The function can be discomposed into three functions: one for the energy level, another for the *active* boards and, finally, for the parameter τ.

$$p_a(s,s') = pn_a((m,n),n') * pm_a(m,m') * p\tau_a(\tau,\tau') \tag{5.2}$$

For the energy harvesting point of view, the transition probability is the same as the one presented in 2.3, $pm_a(m,m') = \delta_{m,m'}$. $pn_a((m,n),n')$ is similar to the one discussed in that section, changing the event probability for the action a:

$$pn_a((m,n),n') = \begin{cases} (1-a)\frac{m}{k} & n' = n+1, 1 \le n < N-1 \\ (1-a)\frac{k-m}{k} + a\frac{m}{k} & n' = n, 1 \le n < N-1 \\ a\frac{k-m}{k} & n' = n-1, 1 \le n \le N-1 \\ \frac{k-m}{k} & n' = n = 0 \\ \frac{m}{k} & n' = 1, n = 0 \\ (1-a) + a\frac{m}{k} & n' = n = N-1 \\ 0 & \text{otherwise} \end{cases} \tag{5.3}$$

Concerning the parameter τ, when the action is not to run an event, then τ increases until its maximum value, T. When the policy is to run an event, if it is successful τ transits to 0, and if not it increases:

$$p\tau_a(\tau,\tau') = \begin{cases} 1-a(1-p_l) & \tau' = \tau+1, 0 \le \tau < T-1 \\ 1-a(1-p_l) & \tau' = \tau = T-1 \\ a(1-p_l) & \tau' = 0 \\ 0 & \text{otherwise} \end{cases} \tag{5.4}$$

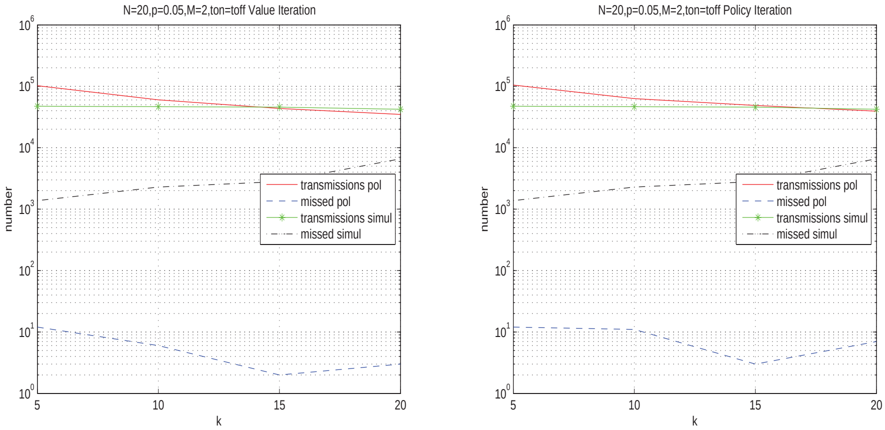


Figure 5.1: Successful and Missed Transmissions $T_{on} = T_{off}$

Reward function

We propose a reward function that increases the probability to run an event when the residual energy available is high compared to the total capacity and decreased when less energy is available. Remember that when there is no enough energy to send a sensing report, that action is not possible. The reward for running an event also increases when the last time the node experienced a successful event when τ is closer to the desired T. In addition, the expected harvested energy is also a factor that the function takes into account. The higher is the energy harvested, the higher is the reward obtained. We define the reward for state s and action a as:

$$r(s,a) = \begin{cases} e^{\frac{n}{N}\sqrt{\frac{\tau+1}{T}} - \frac{k-M}{k} e^{\frac{N-n}{N}\left(\frac{k-m}{k}+p_l\right)}} & a = 1 \\ \frac{k-M}{k} e^{\frac{N-n}{N}\frac{m}{k}} & a = 0 \end{cases} \tag{5.5}$$

Note that the node gets more reward if it runs two events consecutively instead of waiting one time slot and then sending the report.

5.4 Results

For simulating the Markov Decision Processes framework, we used some of the functions implemented in the ”MDP Toolbox v3.0 for MATLAB” by *INRA Toulouse*; [3]. This toolbox incorporates the development of both methods we analyzed, the Value Iteration and the Policy Iteration.

In figures 5.1, 5.2 and 5.3 we observe the number of successful transmissions and the number of missed events, one following a random case simulation and the other following the decisions taken by the corresponding policy. Every figure is composed by two sub-figures, one for the Value Iteration method and one for the Policy Iteration. All the situations share the same parameters except for the average state

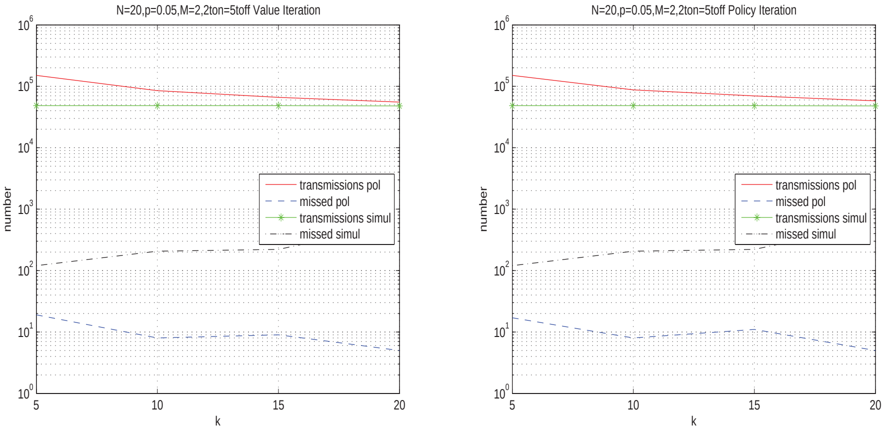


Figure 5.2: Successful and Missed Transmissions $2T_{on} = 5T_{off}$

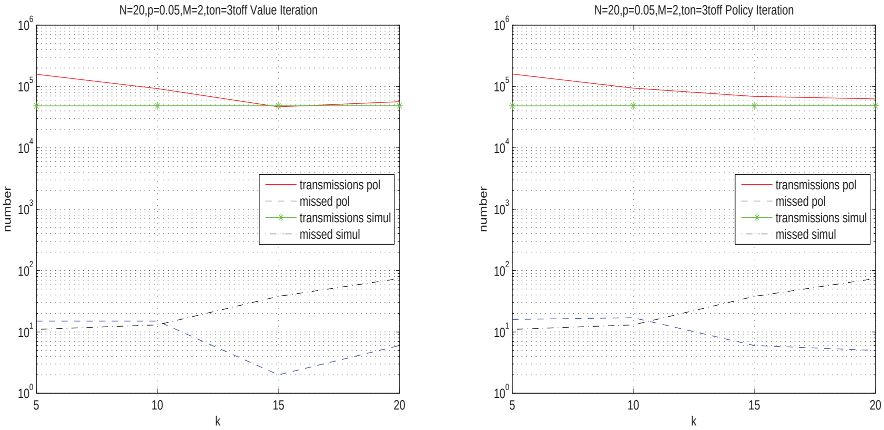


Figure 5.3: Successful and Missed Transmissions $T_{on} = 3T_{off}$

time for the harvesting boards. Those parameters are $N = 20$, $p = 0.05$ and $M = 2$. The average time for the harvesters were $T_{on} = T_{off}$, $2T_{on} = 5T_{off}$ and $T_{on} = 3T_{off}$, respectively. The probability of error in an event transmission is 10^{-4}.

Figure 5.4 shows the percentage of improvement between using the MDP model and the random situation for all the cases and methods. This percentage is only computed considering the number of successful transmission, not the missed events. The MDP cases do not miss any event because of the lack of energy, recall that the action of running an event while being in any state with energy $n = 0$ is not possible. Therefore, they are only caused by the irregularities of the channel. However, we observe that the number of missed events in the random scenarios are generally much higher.

We see that the Policy Iteration method performs slightly better than the Value Iteration. When k

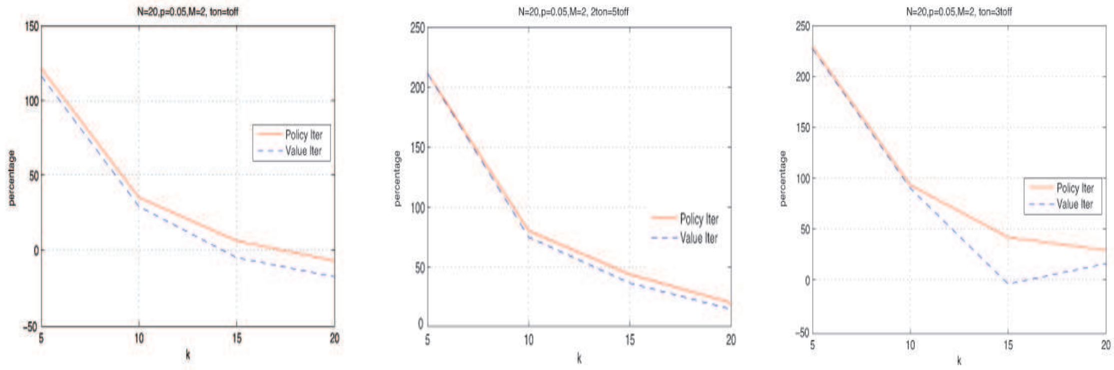


Figure 5.4: Percentage of improvement between mdp and random simulation

is small, the MDP actions showed a great improvement versus the random events, not triggered by the agent's decisions. However, in cases with higher k and lower *active* probabilities of the harvesting boards, the MDP framework is not that useful anymore, and it may even make the node send less sensing reports than the random simulations.

Additionally, in order to computer the optimal actions for each state of the node, it is necessary to compute first the transition probability matrix and then the rewards matrix. The computing time is quite high, that is because we used smaller battery sizes, N, for the simulations.

5.5 Conclusions

In this chapter, we have presented an extension of the MAKERS model, based on the Markov Decision Processes framework. First, we have introduced the concept of the MDP and some of the previous work done on Wireless Sensor Networks. This extended model teaches the node in order to make it able to decide how to react in front of all the possible states, choosing to run an event or wait until the conditions improve, while harvesting energy.

Simulations showed that with good conditions, using the MDP model improves the node performance, increasing their ability to run events up to 230%, for the cases we studied. However, when the harvesting or application conditions are not that favorable, the model does not improve the random case we discussed earlier. Further efforts need to be done on incorporating the consumption for *no-events* time slots and its impact on the reward function proposed. Additionally, real experiments would finally determine the validity of this new approach.

Chapter 6

Open Issues and Conclusions

Wireless Sensor Networks are not only another version of the classical telecommunications networks, but networks with new peculiar requirements. Sensor nodes have limited energy storage capabilities and they necessarily have to harvest energy from their surroundings. This capability has the potential to simultaneously address the conflicting design goals of lifetime and performance. However, precise energy models for these devices are still missing and hence protocol evaluations are not enough accurate.

Therefore, the research community needs to not only cope with an energy model describing the harvesting capabilities of the sensor nodes, but it also have to develop precise energy consumption models that include how they consume the available energy.

This thesis have proposed a Markov Chain based model, called MAKERS, to analyze sensor nodes feed with multiple energy harvesting boards. It captures not only the energy state of the node but also the state of the different energy harvesting boards that feed its energy supply. It allows to calculate the probability of an event, such as a transmission of a sensing report, to be lost because of the lack of energy and, additionally, the average time before the first time this happens depending on the initial energy status.

First, we consider the case of having many boards harvesting from the same energy source. Later, we relaxed this constraint and account for totally different and independent sources. Further, we extended the model adding the power consumption during *no-event* periods.

The simulations and early experiments were done using MATLAB and MICA2 nodes feed with the RF harvesting kit from Powercast. Whereas this was a good an suitable way to compare the differences between the real cases and the model concerning the time before the node runs out of energy, it is not suitable to get any absolute values about the *event-loss* probability. Additionally, the experiments were conducted using only one Energy Harvesting board using RF energy. Hence, the next step would comprise the implementation of better experiments that allow to measure loss probabilities and combine different energy sources within the same sensor node to verify the approach of the model with independent and different energy sources. The results of such tests would provide a good result about the efficiency of the different models and would maybe show a difference to our results on the simulations.

Finally, we also introduced an extension of MAKERS, build upon the concept of Markov Decision Processes. This framework allows the node to decide what operating mode should operate in the next time slot. However, this model has only been described and numerically evaluated, but a reliable simulation analysis and further experiment is still missing. Because of this, we propose to adapt the framework to account for some of the modifications of the first model proposed and evaluate its performance using real sensor nodes, such as the MICA2, using multiple energy harvesting boards.

Bibliography

[1] I.F. Akyildiz, Weilian Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *Communications Magazine, IEEE*, 40(8):102 – 114, August 2002.

[2] R. Bellman. Dynamic programming, 1957.

[3] Biometry and Artificial Intelligence Unit of INRA Toulouse. Markov decision processes (mdp) toolbox v3.0 for matlab. `http://www.inra.fr/internet/Departements/MIA/T/MDPtoolbox/`.

[4] Powercast Corporation. P2000 series 902-928 mhz powerharvester development kit. `http://www.powercastco.com/products/development_kits/`.

[5] L. Kaelbling, M. Littman, and Moore A. Reinforcement learning: A survey. *Journal of Artificial Intelligence Research*, 1996.

[6] A. Kansal, J. Hsu, M. Srivastava, and V. Raqhunathan. Harvesting aware power management for sensor networks. In *Design Automation Conference, 2006 43rd ACM/IEEE*, pages 651 –656, 2006.

[7] A. Kansal, D. Potter, and M. B. Srivastava. *Performance aware tasking for environmentally powered sensor networks*, volume 32. ACM Press, 2004.

[8] Huijiang L., N. Jaggi, and B. Sikdar. Cooperative relay scheduling under partial state information in energy harvesting sensor networks. In *GLOBECOM 2010, 2010 IEEE Global Telecommunications Conference*, dec. 2010.

[9] A. Munir and A. Gordon-Ross. An mdp-based application oriented optimal policy for wireless sensor networks. In *Proceedings of the 7th IEEE/ACM international conference on Hardware/software codesign and system synthesis*, pages 183–192, 2009.

[10] V. Raghunathan, S. Ganeriwal, and M. Srivastava. Emerging techniques for long lived wireless sensor networks. *Communications Magazine, IEEE*, 44(4):108 – 114, April 2006.

[11] A. Seyedi and B. Sikdar. Modeling and analysis of energy harvesting nodes in wireless sensor networks. In *Communication, Control, and Computing, 2008 46th Annual Allerton Conference on*, pages 67 –71, September 2008.

[12] A. Seyedi and B. Sikdar. Performance modeling of transmission schedulers for sensor networks capable of energy harvesting. In *Communications (ICC), 2010 IEEE International Conference on*, may 2010.

[13] Satinder P. Singh. Learning to solve markovian decision processes, 1994.

[14] S. Sudevalayam and P. Kulkarni. Energy harvesting sensor nodes: Survey and implications. *Communications Surveys Tutorials, IEEE*, PP(99):1 –19, 2010.

[15] Richard S. Sutton and Andrew G. Barto. Reinforcement learning i: Introduction, 1998.

[16] Crossbow Technology. Mica2 sensor node datasheet. `http://www.xbow.com/Products/Product_pdf_files`

[17] Berkeley University of California. Tiny os. `http://www.tinyos.net`.

[18] C. Van Phan, K. Baek, and J. Kim. Opportunistic transmission for wireless sensor networks under delay constraints. In *Computational Science and Its Applications ICCSA 2007*. 2007.

[19] R. Williams and L. Baird. Tight performance bounds on greedy policies based on imperfect value functions. In *Tenth Yale Workshop on Adaptive and Learning Systems, Yale University*, June 1994.

[20] Jennifer Yick, Biswanath Mukherjee, and Dipak Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292 – 2330, 2008.

[21] A. Zhang and A. Seyedi. Analysis and design of energy harvesting wireless sensor networks with linear topology. In *Proceedings of the IEEE ICC 2011*, June 2011.